



**TUGAS AKHIR - TE 141599**

## **PERANCANGAN PID-NEURO UNTUK PENGATURAN SUHU AIR PADA PLANT HEAT EXCHANGER**

Arie Imanda  
NRP 2215105045

Dosen Pembimbing  
Ir. Rusdhianto Efendi AK , MT

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**TUGAS AKHIR - TE 141599**

## **PERANCANGAN PID-NEURO UNTUK PENGATURAN SUHU AIR PADA PLANT HEAT EXCHANGER**

Arie Imanda  
NRP 2215105045

Dosen Pembimbing  
Ir. Rusdhianto Efendi AK , MT

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**FINAL PROJECT - TE 141599**

## **PID-NEURO DESIGN FOR WATER TEMPERATURE SETTING ON PLANT HEAT EXCHANGER**

Arie Imanda  
NRP 2215105045

Supervisor  
Ir. Rusdhianto Efendi AK , MT

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Electrical Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul "**Perancangan PID-Neuro Untuk Pengaturan Suhu Air Pada *Plant Heat Exchanger***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan hasil merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan berlaku.

Surabaya, Juli 2017

Arie Imanda

Nrp. 2215 105 045

*Halaman ini sengaja dikosongkan*

**PERANCANGAN PID-NEURO UNTUK PENGATURAN SUHU AIR  
PADA PLANT HEAT EXCHANGER**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

**Bidang Studi Teknik Sistem Pengaturan  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

**Dosen Pembimbing,**

  
**Ir. Rusdhianto Efendi AK, MT**  
**NIP. 195704241985021001**



*Halaman ini sengaja dikosongkan*

## PERANCANGAN PID-NEURO UNTUK PENGATURAN SUHU AIR PADA *PLANT HEAT EXCHANGER*

Nama Mahasiswa : ArieImanda  
Dosen Pembimbing : Ir. RusdhiantoEfendi AK , MT.

### ABSTRAK

*Heat Exchanger* (HE) adalah suatu alat yang memungkinkan perpindahan panas dan bisa berfungsi sebagai pemanas maupun sebagai pendingin. Biasanya, medium pemanas dipakai uap lewat panas (*super heated steam*) dan air biasa sebagai air pendingin (*cooling water*). Permasalahan yang terjadi dalam pengendalian temperatur sistem kerja *heat exchanger* adalah terjadinya perubahan laju *fluida* yang akan menyebabkan terjadinya perubahan parameter-parameter pada *plant heat exchanger* yang dapat mempengaruhi *output* temperatur pada *heat exchanger*. Pada tugas akhir ini dilakukan penelitian tentang penggunaan kontroler PID-*Neural network* pada pengendalian laju debit *steam* pada *heat exchanger* karena *respon output* pada *plant heat exchanger* selalu mengalami keterlambatan, karena proses perpindahan panas sehingga menimbulkan *time delay* pada *respon output*. Simulasi dilakukan menggunakan simulink pada *software* matlab. Simulasi dilakukan dengan cara membandingkan *respon plant* pada saat sebelum dipasang kontroler dan sesudah dipasang kontroler. Simulasi juga dilakukan pada saat *plant* diberi *disturbance* dan tanpa *disturbance* dengan beban konstan dan bervariasi. Hasil dari pengujian dapat dilihat bahwa penerapan kontroler PID-*Neural network* telah mampu beradaptasi terhadap perubahan dari temperatur, dimana *error* yang terjadi pada penelitian ini mencapai 0,1% dan tidak terjadi osilasi pada *respon* sinyal *output* pada *plant heat exchanger*.

Kata kunci: Simulasi *plant*, debit aliran, *heat exchanger*, PID-*Neural network*



*Halaman ini sengaja dikosongkan*

## ***PID-NEURO DESIGN FOR WATER TEMPERATURE SETTING ON PLANT HEAT EXCHANGER***

*Name of Student* : ArieImanda  
*Supervisor* : Ir. RusdhiantoEfendiAK , MT.

### ***ABSTRACT***

*Heat Exchanger (HE) is a tool that allows heat transfer and can serve as a heater or as a coolant. Typically, heating medium is used steam through heat (super heated steam) and water as cooling water (cooling water). Problems that occur in controlling the working temperature of the heat exchanger system is the change of the fluid rate which will cause the change of parameters in the plant heat exchanger which can affect the temperature output at the heat exchanger. In this final project, we conducted a research on the use of PID-Neural network controller on steam discharge rate control on heat exchanger because the output response at heat exchanger plant always experience delay, because heat transfer process causing time delay in output response. Simulation done using simulink in matlab software. The simulation is done by comparing the plant response at the time before the controller is installed and after mounting the controller. Simulation is also done when the plant is given disturbance and without disturbance with constant and variable load. The results of the test can be seen that the application of PID-Neural network controller has been able to adapt to changes in temperature, where the error occurred in this study reach 0.1% and no oscillation in the output signal response at the plant.*

*Keywords: Plant simulation, flow debit, heat exchanger, PID-Neural network*

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “**Perancangan PID-Neuro Untuk Pengaturan Suhu Air Pada *Plant Heat Exchanger***”

Tugas Akhir ini merupakan sebagian syarat untuk menyelesaikan studi dan memperoleh gelar Sarjana Teknik di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada :

1. Kedua orang tua atas limpahan doa, kasih sayang dan teladan hidup bagi penulis.
2. Bapak Ir. Rusdhianto Efendi AK, MT selaku dosen pembimbing.
3. Seluruh staf pengajar dan administrasi Departemen Teknik Elektro FTE-ITS.
4. Teman-teman D3 Computer Control 2k15 khususnya GW25C yaitu: putu, putra, ady, hanif, doni, rizky, arbi, fajar, indra, imam, tantowi, indra, fathur, tegar, calvin, adib, umar, farid, irfan, galuh, abi. yang telah membantu dan menghibur untuk menyelesaikan tugas akhir ini.
5. Cindy fatmawati yang telah memberikan semangat dari awal mengerjakan tugas akhir ini sampai terselesaikannya tugas akhir ini.
6. Semua pihak yang telah banyak membantu untuk menyelesaikan tugas akhir ini yang tidak dapat penulis sebutkan satu persatu.

Harapan kami sebagai penulis adalah semoga terselesaikannya Tugas Akhir ini dapat bermanfaat bagi kami serta pembaca. Sadar atas keterbatasan yang dimiliki oleh penulis karena hasil dari Tugas Akhir ini jauh dari kesempurnaan. Demikian penulis sudah berusaha semaksimal mungkin dan pintu maaf serta saran kritik yang membangun penulis harapkan.

Surabaya, Juli 2017

Penulis

*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

<b>JUDUL .....</b>	<b>i</b>
<b>PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	<b>iii</b>
<b>PENGESAHAN.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
<b>DAFTAR TABEL .....</b>	<b>xvii</b>

<b>BAB I. PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Permasalahan.....	2
1.3. Batasan Masalah.....	2
1.4. Maksud dan Tujuan .....	2
1.5. Metodologi .....	2
1.6. Sistematika Laporan .....	3
1.7. Relevansi.....	4
<b>BAB II. TEORI PENUNJANG .....</b>	<b>5</b>
2.1. <i>Heat Exchanger</i> .....	5
2.1.1 <i>Shell and Tube</i> .....	6
2.1.2 Dimensi <i>Shell and Tube Heat Exchanger</i> .....	7
2.1.3 Tata Letak <i>Shell Side Tube</i> .....	8
2.2 Prinsip Kerja <i>Heat Exchanger (Shell-and-Tube)</i> .....	9
2.3. Mekanisme Perpindahan Panas .....	11
2.3.1. Konduksi .....	11
2.3.2. Konveksi .....	12
2.3.3. Radiasi .....	13
2.4. PID .....	14
2.4.1. Kontrol <i>Proportional</i> .....	14
2.4.2. Kontrol <i>Integratif</i> .....	14
2.4.3. Kontrol <i>Derivatif</i> .....	15
2.5. Jaringan Syaraf Tiruan .....	15
2.5.1. Komponen Jaringan Syaraf Tiruan .....	15
2.5.2. Fungsi Aktivasi.....	15
2.5.3. Proses Pembelajaran .....	19

2.5.4. Jaringan <i>Single Perceptron</i> .....	20
2.5.5. Metode <i>Backpropagation</i> .....	21
2.6. <i>PID-Neural Network</i> .....	25
<b>BAB III. PERANCANGAN SISTEM</b> .....	27
3.1. Diagram Alir Penelitian .....	27
3.2. Identifikasi Kebutuhan Sistem .....	28
3.3. Perancangan dan Pemodelan <i>Heat Exchanger</i> .....	28
3.4. Identifikasi Pemodelan <i>Plant Heat Exchanger</i> .....	33
3.5. Perancangan Kontroler <i>PID-Neural Network</i> .....	34
<b>BAB IV. PENGUJIAN DAN ANALISA DATA</b> .....	41
4.1. <i>Set Point</i> 70 °C dengan Beban Konstan .....	41
4.2. <i>Set Point</i> 70 °C dengan Beban Bervariasi ( <i>Random</i> ) .....	42
4.3. <i>Set Point</i> 90 °C dengan Beban Konstan .....	44
4.4. <i>Set Point</i> 90 °C dengan Beban Bervariasi ( <i>Random</i> ) .....	45
4.5. <i>Set Point</i> 110 °C dengan Beban Konstan .....	46
4.6. <i>Set Point</i> 110 °C dengan Beban Bervariasi ( <i>Random</i> ) .....	47
4.7. <i>Set Point</i> 70 °C dengan Beban Konstan dan <i>Disturbance</i> .....	49
4.8. <i>Set Point</i> 70 °C dengan Beban Bervariasi dan <i>Disturbance</i> .....	50
<b>BAB V. PENUTUP</b> .....	53
5.1. Kesimpulan .....	53
5.2. Saran .....	53
<b>DAFTAR PUSTAKA</b> .....	55
<b>LAMPIRAN</b> .....	56
<i>Listing Program Plant Heat Exchanger</i> .....	A-1
<i>Listing Program Kontroler PID-Neural Network</i> .....	A-2
<b>DAFTAR RIWAYAT HIDUP</b> .....	60

## DAFTAR GAMBAR

Gambar 2.1	Proses pada <i>Heat Exchanger (Shell and Tube)</i> .....	5
Gambar 2.2	<i>Heat Exchanger One-Shell and One-Pass</i> .....	6
Gambar 2.3	Skema Satu-Shell Dua-Pass <i>Shell-and-Tube Heat Exchanger</i> .....	7
Gambar 2.4	Dimensi <i>Shell-and-Tube Heat Exchanger</i> .....	7
Gambar 2.5	<i>Layout</i> Persegi - <i>Layout</i> Belah Ketupat .....	8
Gambar 2.6	<i>Direct He</i> dan <i>Indirect He</i> .....	10
Gambar 2.7	Proses Perpindahan Panas Secara Konduksi .....	12
Gambar 2.8	Perpindahan Panas Konveksi dari Permukaan Media Padat ke <i>Fluida</i> yang Mengalir .....	12
Gambar 2.9	Proses Perpindahan Panas Secara Radiasi .....	13
Gambar 2.10	Struktur <i>Neuron</i> Jaringan Syaraf Tiruan .....	16
Gambar 2.11	Jaringan Syaraf dengan 3 Lapisan. ....	17
Gambar 2.12	Fungsi Aktivasi ( <i>Linear</i> ) Identitas. ....	17
Gambar 2.13	Fungsi Aktivasi ( <i>Saturating Linear</i> ). ....	18
Gambar 2.14	Fungsi Aktivasi <i>Symetric Saturating Linear</i> . ....	18
Gambar 2.15	Arsitektur Jaringan <i>Backpropagation</i> .....	22
Gambar 2.16	Struktur <i>PID-Neural Network</i> .....	25
Gambar 3.1	Diagram Alir Penelitian .....	27
Gambar 3.2	Pembagian Zona <i>Heat Exchanger</i> . ....	28
Gambar 3.3	Blok Diagram <i>Plant Heat Exchanger Nonlinear</i> pada Matlab. ....	32
Gambar 3.4	Tampilan Hasil Simulasi <i>Plant Heat Exchanger Nonlinear</i> pada Matlab. ....	33
Gambar 3.5	Tampilan <i>Block Diagram</i> Pendekatan Arx .....	33
Gambar 3.6	<i>Respon</i> Pendekatan Arx Orde Ke-2. ....	34
Gambar 3.7	Blok Diagram untuk Menentukan Nilai Awal $K_p$ , $K_i$ , Dan $K_d$ .....	35
Gambar 3.8	Blok Diagram Perancangan <i>PID-Neural Network</i> .....	36
Gambar 3.9	<i>Transfer Function</i> Orde 1 dari <i>Plant Heat Exchanger</i> .....	36
Gambar 3.10	Blok Diagram Perancangan <i>PID-Neural Network</i> Untuk Pengaturan Suhu Air pada <i>Plant Heat Exchanger</i> .....	39
Gambar 4.1	Blok Diagram <i>Set Point</i> Konstan Sebesar 70 °C dan Beban Konstan .....	40



Gambar 4.2	Hasil Simulink <i>Set Point</i> Konstan Sebesar 70 °C dan Beban Konstan.....	41
Gambar 4.3	Blok Diagram <i>Set Point</i> Konstan Sebesar 70 °C dan Beban Bervariasi .....	42
Gambar 4.4	Hasil Simulink <i>Set Point</i> Konstan Sebesar 70 °C dan Beban Bervariasi .....	42
Gambar 4.5	Blok Diagram <i>Set Point</i> Konstan Sebesar 90 °C dan Beban Konstan. ....	43
Gambar 4.6	Hasil Simulink <i>Set Point</i> Konstan Sebesar 90 °C dengan Beban Konstan. ....	43
Gambar 4.7	Blok Diagram <i>Set Point</i> Konstan Sebesar 90 °C dengan Beban Bervariasi. ....	44
Gambar 4.8	Hasil Simulink <i>Set Point</i> Konstan Sebesar 90 °C dengan Beban Bervariasi. ....	45
Gambar 4.9	Blok Diagram <i>Set Point</i> Konstan Sebesar 110 °C dan Beban Konstan. ....	45
Gambar 4.10	Hasil Simulink <i>Set Point</i> Konstan Sebesar 110 °C dengan Beban Konstan .....	46
Gambar 4.11	Blok Diagram <i>Set Point</i> Konstan Sebesar 110 °C dengan Beban Bervariasi .....	47
Gambar 4.12	Hasil Simulink <i>Set Point</i> Konstan Sebesar 110 °C dengan Beban Bervariasi. ....	47
Gambar 4.13	Blok Diagram <i>Set Point</i> Konstan Sebesar 70 °C dengan Beban Konstan Dan Terdapat <i>Disturbance</i> pada <i>Plant</i> . ....	48
Gambar 4.14	Hasil Simulink <i>Set Point</i> Konstan Sebesar 70 °C dengan Beban Konstan dan <i>Disturbance</i> pada <i>Plant</i> .....	49
Gambar 4.15	Blok Diagram <i>Set Point</i> Konstan Sebesar 70 °C dengan Beban Bervariasi dan Terdapat <i>Disturbance</i> pada <i>Plant</i> .....	50
Gambar 4.16	Hasil Simulink <i>Set Point</i> Konstan Sebesar 70 °C dengan Beban Bervariasi dan <i>Disturbance</i> pada <i>Plant</i> .....	50

## DAFTAR TABEL

Tabel 2.1	Ringkasan <i>Shell-and-Tube</i> Penukar Panas.....	9
Tabel 3.1	Persamaan Koefisien pada <i>Plant Heat Exchanger</i> .....	31
Tabel 3.2	Nilai Awal $K_p$ , $K_i$ , dan $K_d$ untuk Kontroler PID- <i>Neural Network</i> .....	35
Tabel 3.3	Inisialisai <i>Input</i> pada Pemrograman Kontroler PID- <i>Neural Network</i> .....	37

*Halaman ini sengaja dikosongkan*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Heat exchanger* adalah sebuah alat yang berfungsi untuk mentransfer energi panas (*entalpi*) antara dua atau lebih *fluida*, antara permukaan padat dengan *fluida*, atau antara partikel padat dengan *fluida*, pada suhu yang berbeda serta terjadi kontak *termal*. Lebih lanjut, *heat exchanger* dapat pula berfungsi sebagai alat pembuang panas, alat sterilisasi, *pasteurisasi*, pemisahan campuran, *distilisasi* (pemurnian/*ekstraksi*), pembentukan konsentrat, kristalisasi, atau juga untuk mengontrol sebuah proses *fluida* [1].

Satu bagian terpenting dari *heat exchanger* adalah permukaan kontak panas. Pada permukaan inilah terjadi perpindahan panas dari satu zat ke zat yang lain. Semakin luas bidang kontak total yang dimiliki oleh *heat exchanger* tersebut, maka akan semakin tinggi nilai *efisiensi* perpindahan panasnya. *Heat exchanger* dapat digunakan sebagai pemanas (*regenerator*) maupun sebagai pendingin (*recuperator*) tergantung pada tinjauan perpindahan panas yang terjadi. Dalam kehidupan sehari-hari kita tidak akan terlepas pada alat ini baik dari skala kecil, seperti: AC (*Air Conditioner*) maupun skala besar, seperti *powerplant*. Berdasarkan jumlah *fluida* kerjanya, *heat exchanger* dapat dibagi menjadi dua *fluida*, tiga *fluida* dan N-*fluida* ( $N > 3$ ). Sesuai dengan tinjauannya yang ditinjau merupakan jumlah *fluida* kerjanya saja, namun harus sesuai dengan konsep dasar *heat exchanger*, yaitu harus ada yang bertindak sebagai *fluida* panas dan *fluida* dingin dan untuk jumlah sesuai dengan desain. Pada umumnya terdapat dua *fluida* kerja pada *heat exchanger* baik untuk proses pemanasan, pendinginan, penambahan panas maupun penyerapan panas [2].

Masalah ini terjadi akibat dari terbatasnya area kerja *sensor* dan adanya *baffle* pada *plant heat exchanger*, sehingga terjadi keterlambatan *respon* pada *plant heat exchanger*. Keterlambatan *respon* *plant heat exchanger* jika kontroler dengan menggunakan sistem kontrol PID biasa, tidak mampu mengatasi masalah tersebut. Maka digunakan sistem kontrol PID-*Neural network*. Model kesalahan prediksi dirancang berdasarkan model suatu *plant*. Model *plant* tersebut digunakan untuk memprediksi keluaran dari *plant*, sehingga mengurangi terjadi *error* dan sesuai dengan kriteria yang diinginkan.

## 1.2 Permasalahan

Permasalahan yang akan dibahas dalam tugas akhir ini adalah bagaimana caranya untuk mengatur aliran debit *steam* agar suhu air sesuai dengan yang diinginkan tanpa merubah parameter *heat exchanger*, mendapatkan pemodelan matematika dari *plant heat exchanger* dan mengidentifikasi parameter sistem serta mendesain sistem kontrol PID-*Neural network* dan mensimulasikan agar sinyal *output* mengikuti *signal reference* dari *heat exchanger*.

## 1.3 Batasan Masalah

Batasan masalah dalam perancangan PID-*Neural network* untuk pengaturan suhu air pada *plant heat exchanger* ini adalah :

1. Diasumsi bahwa *plant heat exchanger* terisolasi dengan baik, panas yang tidak penting hilang di sekitarnya.
2. Pemodelan *plant* didapatkan dari paper "*Modelling of A Hot Water Drum and Heat Exchanger Process Control Training Sysytem*" [3].
3. Pemodelan pada *plant heat exchanger* adalah SISO dan sistem kontrol yang digunakan PID-*Neural network*.
4. *Software* yang digunakan adalah matlab dan simulink.

## 1.4 Maksud dan Tujuan

Tujuan dari tugas akhir ini adalah merancang sistem kontrol menggunakan metode PID-*Neural network* pada konfigurasi *plant* SISO serta mensimulasikannya. Dan perancangan kontroler PID-*Neural network* sebagai *tuning* parameter untuk mengatur suhu air pada keluaran *heat exchanger* sesuai dengan parameter yang diinginkan.

## 1.5 Metodologi

Dalam pelaksanaan tugas akhir yang berupa perancangan PID-*Neural network* untuk mengatur suhu air pada *plant heat exchanger* ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

1. Studi pustaka dan survei data sekunder  
Kegiatan ini dilakukan dengan mengidentifikasi masalah yang akan dibahas dan mempelajari buku pedoman dari *e-book* atau jurnal yang menunjang teori atau prinsip dasar *plant heat exchanger*. Selain itu, dalam kegiatan ini juga mencari beberapa

- data sekunder pada *plant heat exchanger*, guna mengetahui parameter *plant heat exchanger* yang akan dikontrol.
2. Pemodelan matematika *heat exchanger*.  
Pada tahapan ini akan dilakukan pemodelan matematika pada *plant heat exchanger* agar bisa menemukan persamaan-persamaan dari *plant heat exchanger* agar bisa dikontrol dan disimulasikan.
  3. Perancangan sistem kontroler *PID-Neural network*  
Perancangan sistem kontrol *PID-Neural network* dilakukan pada *software* matlab dan simulink agar persamaan dari *plant heat exchanger* yang sudah ditemukan bisa dikontrol dan disimulasikan dengan matlab dan simulink dan menghasilkan grafik keluaran yang diinginkan.
  4. Simulasi dan analisis data  
Simulasi ini dimaksudkan untuk melihat hasil grafik keluaran dari *plant heat exchanger* serta melihat hasil dari kontrol *PID-Neural network* apakah sesuai dengan parameter yang diinginkan. Dari pengujian simulasi ini juga dilakukan penganalisaan sistem, untuk memperoleh data-data sistem, seberapa besar nilai *error* sistem. Setelah itu, barulah melakukan penyempurnaan lagi terhadap perancangan kontrol *PID-Neural network* tersebut.
  5. Penyusunan laporan  
Penyusunan laporan dilakukan setelah semua tahap terselesaikan sehingga hasil yang diperoleh dari perancangan sistem kontrol *PID-Neural network* sesuai dengan keluaran *plant heat exchanger* yang diinginkan dan dapat dijelaskan secara rinci dan spesifik sesuai dengan data sekunder yang diperoleh.

## 1.6 SistematikaLaporan

Untuk pembahasan lebih lanjut, laporan tugas akhir ini disusun dengan sistematika sebagai berikut:

### **Bab I :Pendahuluan**

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi, serta relevansi tugas akhir yang dibuat.

### **Bab II : Teori Penunjang**

Menjelaskan teori yang berisi teori-teori penunjang yang dijadikan landasan dan mendukung dalam perencanaan dan simulasi sistem

yang dibuat. Beberapa teori tentang dasar *heat exchanger*, PID, *neural network*, matlab.

### **Bab III :Perancangan Sistem**

Bab ini membahas pemodelan matematika sistem *heat exchanger* dan identifikasi sistem agar bisa disimulasikan. Serta perancangan perangkat lunak (*software*) dan termasuk perancangan PID-*Neural network*.

### **Bab IV :Analisa Data**

Bab ini membahas hasil penelitian dan analisa pada *plant heat exchanger* menggunakan kontrol PID-*Neural network* dan menggunakan *software* matlab serta ditampilkan oleh beberapa simulasi.

### **Bab V :Penutup**

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari tugas akhir ini dan saran-saran dari hasil pembahasan sebelumnya yang sudah diperoleh.

## **1.7 Relevansi**

Dengan adanya perancangan PID-*Neural network* untuk mengatur suhu air pada *plant heat exchanger* diharapkan nantinya akan :

- Dapat mengurangi *error* yang berlebih jika terjadi perpindahan energi pada *plant heat exchanger*.
- Mengetahui *respon* keluaran *heat exchanger* jika dikontrol dengan kontroler PID-*Neural network*.
- Sebagai inovasi *tuning* dalam pengontrolan suhu air pada *plant heat exchanger*.

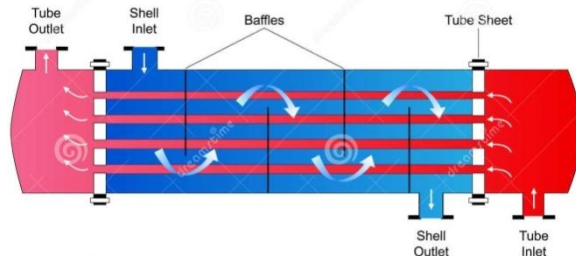
## BAB II

### TEORI PENUNJANG

Dalam bab II, akan dijelaskan mengenai teori dasar dari bahan materi yang dipelajari yang berkaitan dengan perancangan PID-*Neural network* untuk pengaturan suhu air pada plant *heat exchanger*. Antara lain *heat exchanger* dengan tipe *shell and tube*, Pemodelan *nonlinear*, pengaturan *temperature*, kontrol PID-*Neural network*, serta *software* matlab dan simulink untuk melakukan hasil simulasi.

#### 2.1 Heat Exchanger [1]

*Heat Exchanger* (HE) adalah suatu alat yang memungkinkan perpindahan panas dan bisa berfungsi sebagai pemanas maupun sebagai pendingin. Biasanya, medium pemanas dipakai uap lewat panas (*super heated steam*) dan air biasa sebagai air pendingin (*cooling water*). Penukar panas dirancang sebisa mungkin agar perpindahan panas antar *fluida* dapat berlangsung secara *efisien* seperti terlihat pada Gambar2.1.



**Gambar2.1** Proses pada *Heat Exchanger* (*Shell and Tube*).

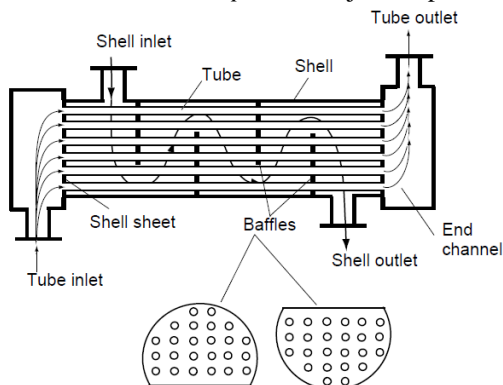
Berdasarkan prinsip kerja *heat exchanger* yang paling efektif adalah mengambil *fluksi* panas (jumlah panas yang berpindah antara dua *fluida*) sebagai *variabel* yang dikontrol, akan tetapi ini tidak mungkin dilakukan mengingat dalam praktiknya *fluksi* panas tersebut sulit diukur.

Pertukaran panas terjadi karena adanya kontak, baik antara *fluida* terdapat dinding yang memisahkannya maupun keduanya bercampur langsung begitu saja. Penukar panas sangat luas dipakai dalam industri seperti kilang minyak, pabrik kimia maupun petrokimia, industri gas alam, refrigerasi, pembangkit listrik.



### 2.1.1 Shell and Tube [1]

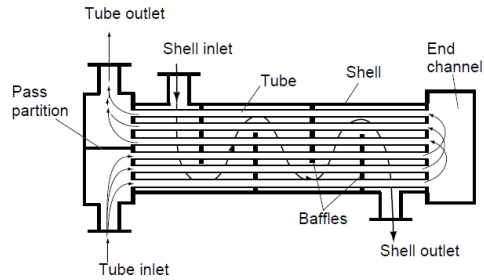
Jenis yang paling umum dari *heat exchanger* dalam proses industri adalah *shell-and-tube heat exchanger*. *Exchanger* menunjukkan lebih dari 65% dari pasar dengan berbagai rancangan dari pengalaman sekitar 100 tahun. *Heat exchanger shell-and-tube* biasanya memberikan kepadatan luas permukaan berkisar antara 50 sampai 500  $m^2/m^3$  dan mudah dibersihkan. Kode desain dan standar tersedia di TEMA (1999) - *Tubular Exchanger Produsen Asosiation*. Sebuah *exchanger* sederhana, yang melibatkan satu *shell* dan satu *pass*, ditunjukkan pada Gambar 2.2.



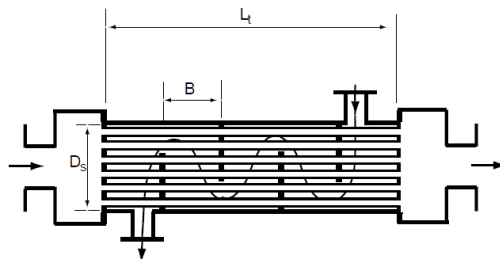
**Gambar 2.2** Heat Exchanger One-Shell and One-Pass.

*Baffles*, Pada Gambar 2.2, *baffle* ditempatkan dalam *shell* dari *heat exchanger* pertama untuk mendukung *tube*, mencegah bergetarnya *tube* dan kendur, dan kedua untuk mengarahkan aliran memiliki koefisien perpindahan panas yang lebih tinggi. Jarak antara dua *baffle* adalah *baffle spacing*.

*Multiple passes*, *heat exchanger shell-and-tube* dapat memiliki beberapa *pass*, seperti 1-1, 1-2, 1-4, 1-6, dan 1-8 *exchangers*, di mana angka pertama menunjukkan jumlah *shell* dan nomor yang kedua menunjukkan jumlah *pass*. Nomor ganjil *pass tube* tidak pernah digunakan kecuali 1-1 *exchanger*. Dan 1-2 *shell-and-tube heat exchanger* diilustrasikan dalam Gambar 2.3.



**Gambar 2.3** Skema Satu-Shell Dua-Pass *Shell-and-Tube Heat Exchanger*.



**Gambar 2.4** Dimensi *Shell-and-Tube Heat Exchanger*.

### 2.1.2 Dimensi *Shell-and-Tube Heat Exchanger* [1]

Beberapa dimensi berikut digambarkan pada Gambar 2.4.

$L$  = Panjang *tube*

$N_t$  = nomer jenis *tube*

$N_p$  = nomer jenis *pass*

$D_s$  = diameter dalam *shell*

$N_b$  = nomer *baffle*

$B$  = jarak *baffle*

Jarak *baffle* diperoleh,

$$B = \frac{L_t}{N_b + 1} \quad (2.1)$$

### 2.1.3 Tata Letak *Shell-Side Tube* [1]

Pada Gambar 2.5 menunjukkan kedua bentuk dari persegi dan *layout* segitiga. *Tube pitch* ( $P_t$ ) dan *clearance* ( $C_t$ ) antara *tube* yang berdekatan keduanya didefinisikan. Persamaan dari diameter *ekivalen* ditulis ,

$$D_e = \frac{4 A_c}{P_{heated}} \quad (2.2)$$

Dari Gambar 2.5, diameter *ekivalen* untuk *layout* persegi adalah

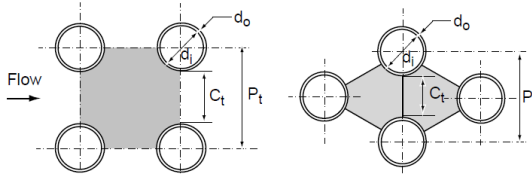
$$D_e = \frac{4(P_t^2 - \pi d_o^2/4)}{\pi d_o} \quad (2.3)$$

Dari Gambar 2.5, diameter *ekivalen* untuk *layout* segitiga adalah

$$D_e = \frac{4 \left( \frac{\sqrt{3} P_t^2}{4} - \frac{\pi d_o^2}{8} \right)}{\pi d_o/2} \quad (2.4)$$

Daerah *cross flow* dari *shell*  $A_c$  didefinisikan sebagai berikut:

$$A_c = \frac{D_s C_t B}{P_t} \quad (2.5)$$



**Gambar 2.5** *Layout* Persegi - *Layout* Belah Ketupat.

Rasio diameter  $d_r$  didefinisikan oleh

$$d_r = \frac{d_o}{d_t} \quad (2.6)$$

Rasio *tube pitch*  $P_r$  didefinisikan oleh

$$P_r = \frac{P_t}{d_o} \quad (2.7)$$

*Tube clearance*  $C_t$  diperoleh,

$$C_t = P_t - d_o \quad (2.8)$$

nomer *tube*  $N_t$  dapat diprediksi pendekatan dengan diameter *shell* dalam  $D_s$  sebagai berikut:

$$N_t = (CTP) \frac{\pi D_s^2/4}{Shade Area} \quad (2.9)$$

Dimana  $CTP$  adalah jumlah *tube* konstan untuk melengkapi cakupan lengkap dari diameter *shell* dengan *tube*, karena *clereance* yang diperlukan antara *shell* dan terluar lingkaran *tube* dan kedalaman *tube* karena jalur *tube pass* untuk beberapa desain *pass*.

$CTP = 0,93$  untuk *one – pass exchanger*

$CTP = 0,9$  untuk *two – pass exchanger*

$CTP = 0,85$  untuk *three – pass exchanger*

$$Shade Area = CL \cdot P_t^2 \quad (2.10)$$

Dimana  $CL$  adalah tata letak tabung konstan.

$CL =$  untuk tata letak *square – pitch*

$CL = \sin(60^\circ) = 0,866$  untuk tata letak *triangular – pitch*

Memasukkan Persamaan (2.10) ke (2.9) memberikan persamaan sebagai berikut:

$$N_t = \frac{\pi}{4} \left( \frac{CTP}{CL} \right) \frac{D_s^2}{P_t^2} = \frac{\pi}{4} \left( \frac{CTP}{CL} \right) \frac{D_s^2}{P_r^2 d_0^2} \quad (2.11)$$

**Tabel 2.1** Ringkasan *Shell-and-Tube* Penukar Panas.

Deskripsi	Persamaan
Persamaan dasar	$q = \dot{m}_1 C_{p1} (T_{1i} - T_{1o})$ $q = \dot{m}_2 C_{p2} (T_{2i} - T_{2o})$
Area transfer panas dari permukaan dalam dan luar dari pipa	$A_i = \pi \cdot d_i \cdot N_t \cdot L$ $A_o = \pi \cdot d_o \cdot N_t \cdot L$
Secara keseluruhan <i>heat transfer Koefisien</i>	$U_o = \frac{1/A_o}{\frac{1}{h_i A_i} + \frac{\ln(d_o/d_i)}{2\pi k L} + \frac{1}{h_o A_o}}$

## 2.2 Prinsip Kerja *Heat Exchanger (Shell and Tube)* [2]

*Heat exchanger* atau sering kita sebut alat penukar panas merupakan alat yang berfungsi untuk memindahkan energi panas antara dua atau lebih *fluida* dan terjadi pada temperatur yang berbeda antara *fluida*, dimana *fluida* tersebut ada yang bertindak sebagai *fluida* panas (*hot fluid*) dan yang lain bertindak sebagai *fluida* dingin (*cold fluid*).

*Heat exchanger* dapat digunakan sebagai pemanas (*regenerator*) maupun sebagai pendingin (*recuperator*) tergantung pada tinjauan perpindahan panas yang terjadi. Dalam kehidupan sehari-hari kita tidak akan terlepas pada alat ini baik dari skala kecil, seperti: AC (*Air Conditioner*) maupun skala besar, seperti: *powerplant*. Jenis *heat*

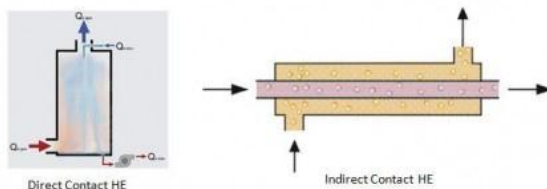
*exchanger* sangat bervariasi dan dapat diklasifikasikan berdasarkan beberapa tinjauan antara lain:

- Proses perpindahan
- Jumlah *fluida* kerja
- Desain konstruksi
- Bidang kontak perpindahan panas
- Arah aliran *fluida*

Untuk penjelasan lebih detail terkait jenis-jenis *heat exchanger* berdasarkan tinjauan di atas mari kita lihat penjelasan berikut.

1. Berdasarkan proses perpindahannya

*Heat exchanger* dapat dibagi menjadi dua, yaitu: *direct contact* dan *indirect contact*. *Direct contact heat exchanger* merupakan *heat exchanger* dimana perpindahan panas antara *fluida* panas dan *fluida* dingin langsung terjadi kontak atau tanpa ada pembatas. Sebaliknya untuk *Indirect heat exchanger*, perpindahan panas antara kedua *fluida* dibatasi oleh suatu dinding pembatas.



**Gambar 2.6** *Direct HE* dan *Indirect HE*.

2. Berdasarkan jumlah *fluida* kerjanya,

*Heat exchanger* dapat dibagi menjadi dua *fluida*, tiga *fluida* dan N-*fluida* ( $N > 3$ ). Sesuai dengan tinjauannya yang ditinjau merupakan jumlah *fluida* kerjanya saja, namun harus sesuai dengan konsep dasar *heat exchanger*, yaitu harus ada yang bertindak sebagai *fluida* panas dan *fluida* dingin dan untuk jumlah sesuai dengan desain. Pada umumnya terdapat dua *fluida* kerja pada *heat exchanger* baik untuk proses pemanasan, pendinginan, penambahan panas maupun penyerapan panas. Untuk penggunaan *fluida* kerja yang lebih dari dua *fluida* biasanya diaplikasikan pada industri yang menggunakan proses kimia seperti proses penghilangan kandungan nitrogen dari bahan baku gas alam.

### 3. Berdasarkan aliran *fluida*

Ditinjau dari aliran *fluida* yang mengalir di dalam *heat exchanger*, *heat exchanger* dapat dikelompokkan menjadi *single pass* dan *multi pass heat exchanger*. *Heat exchanger* dapat disebut *single pass heat exchanger* ketika suatu *fluida* tepat mengalir hanya satu kali di dalam *heat exchanger* tersebut. Sedangkan dikatakan *multi-pass* apabila *fluida* mengalir lebih dari satu kali di dalam sebuah *heat exchanger*. Untuk *single pass heat exchanger* terdapat dua jenis arah alirannya itu *counter flow* dan *parallel flow*.

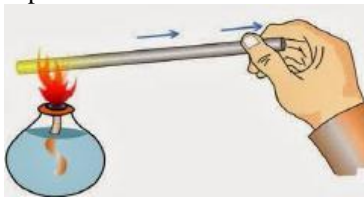
## 2.3 Mekanisme Perpindahan Panas [2]

Pada *heat exchanger* kemampuan untuk menerima panas dipengaruhi oleh 3 hal :

- Koefisien *overall* perpindahan panas ( $U$ )  
Menyatakan mudah atau tidaknya panas berpindah dari *fluida* panas ke *fluida* dingin dan juga menyatakan aliran panas menyeluruh sebagai gabungan proses konduksi dan konveksi.
- Luas bidang yang tegak lurus terhadap arah perpindahan panas.  
Karena luas perpindahan panas tidak konstan, sehingga dalam praktek dipilih luas perpindahan panas berdasarkan luas dinding bagian luar.
- Selisih temperatur rata-rata logaritmik ( $\Delta T_{LMTD}$ ).  
 $LMTD$  : perbedaan temperatur yang dipukul rata-rata setiap bagian HE (*heat exchanger*). Karena perbedaan temperatur di setiap bagian HE tidak sama.

### 2.3.1 Konduksi [2]

Adalah proses perpindahan panas jika panas mengalir dari tempat yang suhunya tinggi ke tempat yang suhunya lebih rendah, dengan media penghantar panas tetap.



**Gambar 2.7** Proses Perpindahan Panas Secara Konduksi.

Persamaan laju konduksi dikenal dengan Hukum *Fourier* (*Fourier Law of Heat Conduction*) tentang konduksi, yang persamaannya matematikanya dituliskan sebagai berikut:

$$Q = -KA \frac{dT}{dx} \quad (2.12)$$

Dimana:

$Q$  = Laju perpindahan panas konduksi (W)

$K$  = Konduktivitas thermal bahan (W/m.K)

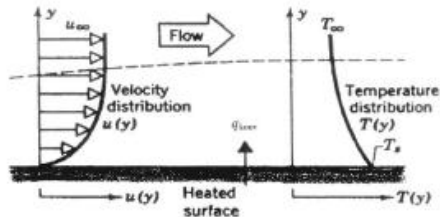
$A$  = Luas penampang tegak lurus terhadap arah aliran panas (m<sup>2</sup>)

$\frac{dT}{dx}$  = Gradien temperatur pada penampang tersebut (K/m)

Tanda (-) diselipkan agar memenuhi hukum Thermodinamika II, yang menyebutkan bahwa, panas dari media bertemperatur lebih tinggi akan bergerak menuju media yang bertemperatur lebih rendah.

### 2.3.2 Konveksi [2]

Adalah proses perpindahan panas jika panas mengalir dari tempat yang suhunya tinggi ke tempat yang suhunya lebih rendah, dengan media penghantar panas tetap.



**Gambar 2.8** Perpindahan Panas Konveksi dari Permukaan Media Padat ke *Fluida* yang Mengalir.

Kalor yang dipindahkan secara konveksi dinyatakan dengan persamaan *newton*, yakni :

$$Q = h A \Delta T \quad (2.13)$$

Dimana :

$Q$  = Kalor yang dipindahkan

$h$  = Koefisien perpindahan panas secara konveksi

$A$  = Luas bidang permukaan perpindahan panas

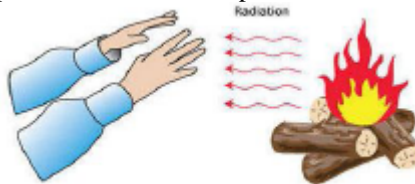
$\Delta T$  = Perubahan temperatur

1. Konveksi bebas/konveksi alamiah (*free convection/natural convection*), perpindahan panas yang disebabkan oleh beda suhu dan beda rapat saja dan tidak ada tenaga dari luar yang mendorongnya. Contoh: plat panas dibiarkan berada di udara sekitar tanpa ada sumber gerakan dari luar.

2. Konveksi paksaan (*forced convection*), perpindahan panas aliran gas atau cairan yang disebabkan adanya tenaga dari luar. Contoh : plat panas dihembus udara dengan kipas/*blower*.

### 2.3.3 Radiasi [2]

Adalah perpindahan panas radiasi dapat dikatakan sebagai proses perpindahan panas dari satu media ke media lain akibat perbedaan *temperature* tanpa memerlukan media perantara.



**Gambar 2.9** Proses Perpindahan Panas Secara Radiasi.

Peristiwa radiasi akan lebih efektif terjadi pada ruang hampa, berbeda dari perpindahan panas konduksi dan konveksi yang mengharuskan adanya media perpindahan panas.

Besarnya radiasi yang dipancarkan oleh permukaan suatu benda nyata (*real*) ( $Q_{rad.g}$ ), adalah:

$$Q_{rad.g} = \varepsilon \cdot \sigma \cdot T_s^4 \cdot A \quad (2.14)$$

Sedangkan, untuk benda hitam sempurna (*black body*), dengan nilai emisivitas ( $\varepsilon = 1$ ) memancarkan radiasi ( $Q_{rad.b}$ ), sebesar:

$$Q_{rad.g} = \sigma \cdot T_s^4 \cdot A \quad (2.15)$$

Untuk laju pertukaran panas radiasi keseluruhan, antara permukaan dengan sekelilingnya (*surrounding*) dengan temperatur sekeliling ( $T_{sur}$ ), adalah:

$$Q_{rad} = \varepsilon \cdot \sigma \cdot (T_s^4 - T_{sur}^4) \cdot A \quad (2.16)$$

Dimana:

$Q_{rad}$  = Laju pertukaran panas radiasi (W)

$\varepsilon$  = Nilai emisivitas suatu benda ( $0 \leq \varepsilon \leq 1$ )



$\sigma$  = Konstanta proporsionalitas, disebut juga konstanta *Stefan Boltzmann*. Dengan nilai  $5,67 \times 10^{-8} (W/m^2 K^4)$

$A$  = Luas bidang permukaan ( $m^2$ )

$T_s$  = Temperatur benda (K)

Dalam hal ini semua analisis tentang temperatur dalam pertukaran panas radiasi adalah dalam temperatur mutlak (*absolut*) yaitu Kelvin (K).

## 2.4 PID [3]

Sistem kontrol PID (*Proportional–Integral–Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut (*Feedback*). Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan

### 2.4.1 Kontrol *Proportional* [3]

Kontrol P jika  $G(s) = K_p$ , dengan  $K$  adalah konstanta. Jika  $u = G(s) \cdot e$  maka  $u = K_p \cdot e$  dengan  $K_p$  adalah konstanta *proporsional*.  $K_p$  berlaku sebagai *gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki *respon transien* khususnya *rise time* dan *settling time*..

### 2.4.2 Kontrol *Integratif* [3]

Jika  $G(s)$  adalah kontrol I maka  $u$  dapat dinyatakan sebagai  $u(t) = [Integral(t)dt]K_i$  dengan  $K_i$  adalah konstanta *integral*, dan dari persamaan diatas,  $G(s)$  dapat dinyatakan sebagai  $u = K_d [\frac{\Delta e}{\Delta t}]$  Jika  $e(t)$  mendekati konstan (bukan nol) maka  $u(t)$  akan menjadi sangat besar sehingga diharapkan dapat memperbaiki *error*. Jika  $e(t)$  mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki

sekaligus menghilangkan *respon steady state*, namun pemilihan  $K_i$  yang tidak tepat dapat menyebabkan *respon transien* yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan  $K_i$  yang sangat tinggi justru dapat menyebabkan *output* berosilasi karena menambah orde sistem.

#### 2.4.3 Kontrol Derivatif [3]

Sinyal kontrol  $u$  yang dihasilkan oleh kontrol D dapat dinyatakan sebagai  $G(s) = s \cdot K_d$ , nampak bahwa sifat dari kontrol D ini dalam konteks "kecepatan" atau *rate* dari *error*. Dengan sifat ini ia dapat digunakan untuk memperbaiki *respon transien* dengan memprediksi *error* yang akan terjadi. Kontrol *derivative* hanya berubah saat ada perubahan *error* sehingga saat *error* statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler *derivative* tidak dapat dipakai sendiri.

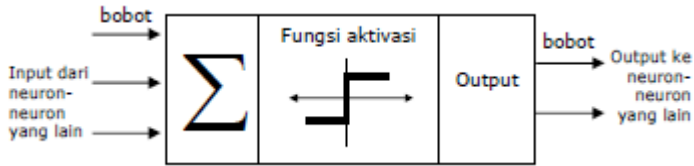
### 2.5 Jaringan Syaraf Tiruan [4]

Kecerdasan buatan (*artificial intelligen* atau disingkat AI) merupakan suatu aplikasi yang meniru kecerdasan manusia untuk digunakan dalam mesin-mesin atau peralatan-peralatan cerdas. konon katanya otak manusia memiliki 2 fungsi utama, yaitu fungsi berpikir dan fungsi belajar. fungsi berpikir manusia tercermin dari kemampuannya untuk berlogika, sedangkan fungsi belajar membuat manusia bisa mengingat sesuatu melalui pembentukan pola-pola di otaknya. Sejalan dengan fungsi otak manusia tersebut, maka AI juga dibagi menjadi 2 kelompok, yaitu AI yang meniru fungsi berpikir dan AI yang meniru fungsi belajar. Salah satu contoh AI yang meniru fungsi berpikir adalah logika *fuzzy* (*fuzzy logic*), sedangkan contoh AI yang meniru fungsi belajar adalah *Neural Network* (NN).

#### 2.5.1 Komponen Jaringan Syaraf Tiruan [4]

Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri-dari beberapa *neuron*, dan ada hubungan antara *neuron-neuron* tersebut. *Neuron-neuron* tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke *neuron-neuron* yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan

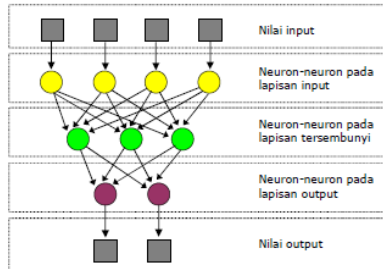
pada suatu nilai tertentu pada bobot tersebut. Gambar 2.10 menunjukkan struktur *neuron* pada jaringan syaraf.



**Gambar 2.10** Struktur *Neuron* Jaringan Syaraf Tiruan.

Jika kita lihat, *neuron* buatan ini sebenarnya mirip dengan sel *neuron* biologis. *Neuron-neuron* buatan tersebut bekerja dengan cara yang sama pula dengan *neuron-neuron* biologis. Informasi (disebut dengan: *input*) akan dikirim ke *neuron* dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron*. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka *neuron* tersebut akan diaktifkan, tapi kalau tidak, maka *neuron* tersebut tidak akan diaktifkan. Apabila *neuron* tersebut diaktifkan, maka *neuron* tersebut akan mengirimkan *output* melalui bobot-bobot *output*-nya ke semua *neuron* yang berhubungan dengannya.

Demikian seterusnya pada jaringan syaraf, *neuron-neuron* akan dikumpulkan dalam lapisan-lapisan(*layer*) yang disebut dengan lapisan *neuron* (*neuron layers*). Biasanya *neuron-neuron* pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan. Gambar 2.11 menunjukkan jaringan syaraf dengan 3 lapisan.



**Gambar 2.11** Jaringan Syaraf dengan 3 Lapisan.

Gambar 2.11 bukanlah struktur umum jaringan syaraf. Beberapa jaringan syaraf ada juga yang tidak memiliki lapisan tersembunyi, dan ada juga jaringan syaraf dimana *neuron-neuron* disusun dalam bentuk matriks.

### 2.5.2 Fungsi Aktivasi [4]

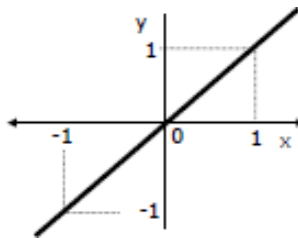
Mengaktifkan jaringan syaraf tiruan berarti mengaktifkan setiap *neuron* yang dipakai pada jaringan tersebut. Banyak fungsi yang dapat dipakai sebagai pengaktif, seperti fungsi *unit step*, *impulse*, *sigmoid*, dan lain-lain. Pengguna fungsi *sigmoid* dianggap sebagai fungsi yang lebih mendekati dengan kinerja otak. Sedangkan fungsi yang akan dipakai pada perancangan ini adalah fungsi *linear*.

Ada beberapa fungsi aktivasi *linear* yang sering digunakan dalam jaringan syaraf tiruan, antara lain:

a) Fungsi *linear* (identitas)

Fungsi linear memiliki nilai *output* yang sama dengan nilai *input*-nya (Gambar 8.11). Fungsi *linear* dirumuskan sebagai berikut:

$$y = x \quad (2.16)$$

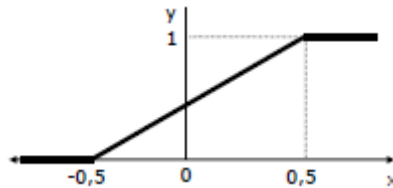


**Gambar 2.12** Fungsi Aktivasi (*Linear*) Identitas.

b) Fungsi *saturating linear*

Fungsi ini akan bernilai 0 jika *input*-nya kurang dari  $-\frac{1}{2}$ , dan akan bernilai 1 jika *input*-nya lebih dari  $\frac{1}{2}$ . Sedangkan jika nilai *input* terletak antara  $-\frac{1}{2}$  dan  $\frac{1}{2}$ , maka *output*-nya akan bernilai sama dengan nilai *input* ditambah  $\frac{1}{2}$  (Gambar 2.13). Fungsi *saturating linear* dirumuskan sebagai berikut:

$$y = \begin{cases} 1, & \text{jika } x \geq 0,5 \\ x + 0,5, & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases} \quad (2.17)$$

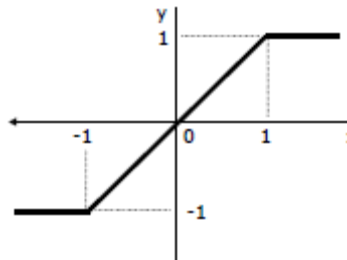


**Gambar 2.13** Fungsi Aktivasi (*Saturating Linear*).

c) Fungsi *symetric saturating linear*

Fungsi ini akan bernilai -1 jika *input*-nya kurang dari  $-1$ , dan akan bernilai 1 jika *input*-nya lebih dari 1. Sedangkan jika nilai *input* terletak antara  $-1$  dan 1, maka *output*-nya akan bernilai sama dengan nilai *input*-nya (Gambar 2.14). Fungsi *symetric saturating linear* dirumuskan sebagai berikut:

$$y = \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ 0; & \text{jika } x \leq -1 \end{cases} \quad (2.18)$$



**Gambar 2.14** Fungsi Aktivasi *Symetric Saturating Linear*.

### 2.5.3 Proses Pembelajaran [4]

Pada otak manusia, informasi yang dilewatkan dari satu *neuron* ke *neuron* yang lainnya berbentuk rangsangan listrik melalui dendrit. Jika rangsangan tersebut diterima oleh suatu *neuron*, maka *neuron* tersebut akan membangkitkan *output* ke semua *neuron* yang berhubungan dengannya sampai informasi tersebut sampai ke tujuannya yaitu terjadinya suatu reaksi. Jika rangsangan yang diterima terlalu halus, maka *output* yang dibangkitkan oleh *neuron* tersebut tidak akan di *respon*. Tentu saja sangatlah sulit untuk memahami bagaimana otak manusia bisa belajar. Selama proses pembelajaran, terjadi perubahan yang cukup berarti pada bobot-bobot yang menghubungkan antar *neuron*. Apabila ada rangsangan yang sama dengan rangsangan yang telah diterima oleh *neuron*, maka *neuron* akan memberikan reaksi dengan cepat. Namun apabila kelak ada rangsangan yang berbeda dengan apa yang telah diterima oleh *neuron*, maka *neuron* akan segera beradaptasi untuk memberikan reaksi yang sesuai.

Jaringan syaraf akan mencoba untuk mensimulasikan kemampuan otak manusia untuk belajar. Jaringan syaraf tiruan juga tersusun atas *neuron-neuron* dan *dendrit*. Tidak seperti model biologis, jaringan syaraf memiliki struktur yang tidak dapat diubah, dibangun oleh sejumlah *neuron*, dan memiliki nilai tertentu yang menunjukkan seberapa besar koneksi antara *neuron* (yang dikenal dengan nama bobot). Perubahan yang terjadi selama proses pembelajaran adalah perubahan nilai bobot. Nilai bobot akan bertambah, jika informasi yang diberikan oleh *neuron* yang bersangkutan tersampaikan, sebaliknya jika informasi tidak disampaikan oleh suatu *neuron* ke *neuron* yang lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi. Pada saat pembelajaran dilakukan pada *input* yang berbeda, maka nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Apabila nilai ini telah tercapai mengindikasikan bahwa tiap-tiap *input* telah berhubungan dengan *output* yang diharapkan.

#### a) Pembelajaran terawasi (*supervised learning*)

Metode pembelajaran pada jaringan syaraf disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Contoh: andaikan kita memiliki jaringan syaraf yang akan digunakan untuk mengenali pasangan pola, misalkan pada operasi AND:

<i>Input</i>	<i>Target</i>
0 0	0
0 1	0
1 0	0
1 1	1

Pada proses pembelajaran, satu pola *input* akan diberikan ke satu *neuron* pada lapisan *input*. Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke *neuron* pada lapisan *output*. Lapisan *output* ini akan membangkitkan pola *output* yang nantinya akan dicocokkan dengan pola *output target*-nya. Apabila terjadi perbedaan antara pola *output* hasil pembelajaran dengan pola *target*, maka disini akan muncul *error*. Apabila nilai *error* ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi.

b) Pembelajaran tak terawasi (*unsupervised learning*)

Pada metode pembelajaran yang tak terawasi ini tidak memerlukan *target output*. Pada metode ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu *area* tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola.

#### 2.5.4 Jaringan *Single Perceptron* [5]

Jaringan lapis tunggal *perceptron* (*single layer perceptron*) terdiri dari beberapa unit pemroses (*neuron*), yang terhubung dan mempunyai beberapa *input* serta sebuah atau beberapa *output*. *Single perceptron* sering disebut juga dengan *perceptron*. *Perceptron* menghitung jumlah nilai perkalian pembobot dan *input* dari parameter permasalahan yang kemudian dibandingkan dengan nilai *target*. Bilai nilai *output* lebih besar atau lebih kecil dari *target*, akan dijadikan sebagai *error*.

Secara matematis perkalian antara pembobot dan parameter *input* dapat dituliskan sebagai berikut:

$$\text{input} = w_{ji}x_i \quad (2.19)$$

Dimana  $x_i$  , adalah sinyal *input* dan  $w_{ji}$  , adalah pembobot. pelatihan pada *perceptron* ini, satu siklusnya diakhiri dengan pengubahan nilai

pembobotnya hingga sejumlah pelatihan yang dilakukan dengan membandingkan *output* dari jaringan dengan *targetnya* dan proses tersebut dituliskan seperti berikut ini:

$$w_{baru} = w_{lama} + \alpha(t_j - O_j)x_i \quad (2.20)$$

### 2.5.5 Metode *Backpropagation* [5]

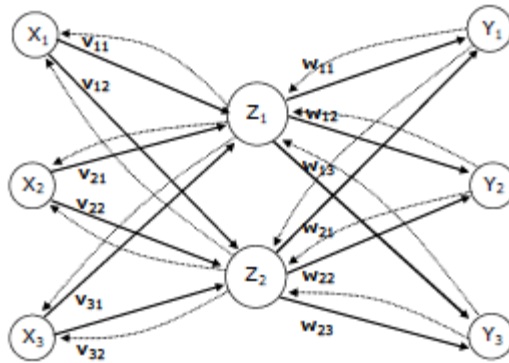
Algoritma pelatihan *backpropagasi* (*backpropagation*) atau ada yang menerjemahkannya menjadi *propagasi balik*, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh *Rumelhart* dan *McClelland* untuk dipakai pada JST, dan selanjutnya algoritma ini biasa disingkat BP. Algoritma ini termasuk metode pelatihan *supervised* dan didesain untuk operasi pada jaringan *feed forward* multi lapis.

Metode BP ini banyak diaplikasikan secara luas. Sekitar 90%, bahkan lebih BP telah berhasil diaplikasikan diberbagai bidang, diantaranya diterapkan di bidang finansial, pengenalan pola tulisan tangan, sistem kendali, pengolahan citra medika dan masih banyak lagi keberhasilan BP sebagai salah satu metode komputasi yang handal.

Algoritma ini juga banyak dipakai pada aplikasi pengaturan karena proses pelatihannya didasarkan pada hubungan yang sederhana, yaitu: jika *output* memberikan hasil yang salah, maka pembobot (*weight*) dikoreksi supaya *error*-nya dapat diperkecil dan *respon* jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar. BP juga berkemampuan untuk memperbaiki pembobot pada lapisan tersembunyi (*hidden layer*).

Secara garis besar, alasan algoritma ini disebut sebagai *propagasi balik*, dapat dideskripsikan sebagai berikut: ketika jaringan diberikan pola *input* sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit-unit lapisan *output*. Kemudian unit-unit lapisan *output* memberikan tanggapan yang disebut sebagai *output* jaringan. Saat *output* jaringan tidak sama dengan *output* yang diharapkan maka *output* akan menyebar mundur (*backward*) pada lapisan tersembunyi diteruskan ke unit pada lapisan *input*. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *backpropagation* atau *propagasi balik*. Arsitektur jaringan *backpropagation* seperti terlihat pada Gambar 2.15.





**Gambar 2.15** Arsitektur Jaringan *Backpropagation*.

Tahap pelatihan ini merupakan langkah bagaimana suatu jaringan syaraf itu berlatih, yaitu dengan cara melakukan perubahan pembobot (sambungan antar lapisan yang membentuk jaringan melalui masing-masing unitnya). Sedangkan pemecahan masalah baru akan dilakukan jika proses pelatihan tersebut selesai. Fase tersebut adalah fase *mapping* atau proses pengujian/*testing*. Algoritma pelatihan *backpropagation* terdiri dari dua proses, *feed forward* dan *backpropagation* dari *error*-nya. untuk jelasnya dapat dijelaskan rinciannya sebagai berikut:

**Langkah 1:**

Pemberian inisialisasi pembobot (diberi nilai kecil secara acak).

**Langkah 2:**

Untuk masing-masing pasangan data pelatihan (*training data*) lakukan langkah 2 sampai 7. Untuk sistem pembelajaran saat propagasi maju (*feed forward*), maka digunakan langkah-langkah sebagai berikut:

**Langkah 3:**

Masing-masing unit *input* ( $x_i, i = 1, \dots, n$ ) menerima sinyal *input*  $x_i$  dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit-unit lapisan tersembunyi).

**Langkah 4:**

Masing-masing unit dilapisan tersembunyi dikalikan dengan faktor pembobot dan dijumlahkan serta ditambah dengan biasnya:

$$Zin_j = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (2.21)$$

Kemudian menghitung sesuai dengan fungsi aktivasi yang digunakan:

$$Z_j = f(Zin_j) \quad (2.22)$$

### Langkah 5:

Masing-masing unit *output* ( $Y_k, k = 1, 2, 3, \dots, m$ ) dikalikan dengan faktor pembobot dan dijumlahkan:

$$Yin_k = W_{ok} + \sum_{j=1}^p Z_j W_{jk} \quad (2.23)$$

Hitung kembali aktivasi

$$yk = f(yin_k) \quad (2.24)$$

Untuk sistem pembelajaran saat propagasi balik (*backpropagation*) sekaligus menghitung nilai *error* yang ada adalah:

### Langkah 6:

Masing-masing unit *output* ( $Y_k, k = 1, \dots, m$ ) menerima pola *target* sesuai dengan pola *input* saat pelatihan atau *training* dan dihitung *error*-nya:

$$\delta_k = (t_k - y_k) f'(yin_k) \quad (2.25)$$

Menghitung perbaikan faktor pembobot (kemudian untuk memperbaiki  $w_{jk}$ ):

$$\Delta W_{kj} = \alpha \cdot \delta_k \cdot Z_j \quad (2.26)$$

Menghitung perbaikan bias koreksi bias:

$$\Delta W_{ok} = \alpha \cdot \delta_k \quad (2.27)$$

dan menggunakan nilainya pada semua unit lapisan sebelumnya.

### Langkah 7:

Masing-masing pembobot yang menghubungkan unit-unit lapisan *output* dengan unit-unit pada lapisan tersembunyi ( $Z_j, j = 1, \dots, p$ ) dikalikan delta dan dijumlahkan sebagai *input* ke unit-unit lapisan berikutnya.

$$\delta in_j = \sum_{k=1}^m \delta_k W_{jk} \quad (2.28)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasinya untuk menghitung *error*-nya.

$$\delta_j = \delta in_j f'(yin_j) \quad (2.29)$$

Kemudian menghitung perbaikan pembobot (digunakan untuk memperbaiki  $V_{ij}$ ).

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (2.30)$$

Kemudian menghitung perbaikan bias (untuk memperbaiki  $V_{oj}$ )

$$\Delta V_{oj} = \alpha \delta_j \quad (2.31)$$

**Langkah 8:**

Masing-masing *output* unit ( $y_k$ ,  $k=1, \dots, m$ ) diperbaiki bias dan pembobotnya ( $j=0, \dots, p$ )

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.32)$$

Masing-masing unit tersembunyi ( $Z_j$ ,  $j=1, \dots, p$ ) diperbaiki bias dan pembobotnya ( $j=0, \dots, n$ ).

$$V_{jk}(\text{baru}) = V_{jk}(\text{lama}) + \Delta V_{jk} \quad (2.33)$$

**Langkah 9:**

Uji kondisi pemberhentian (akhir iterasi).

**Langkah 10:**

Ulangi langkah 1 hingga 8 sampai kondisi akhir iterasi dipenuhi.

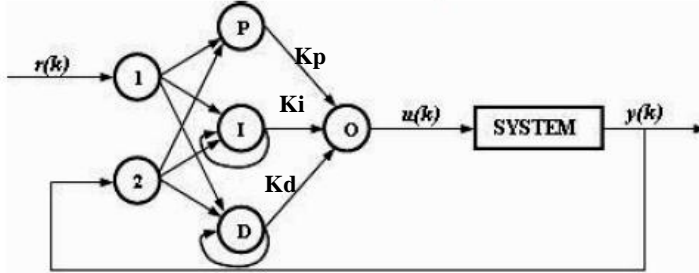
**Daftar Notasi**

$x$	: Pola <i>input</i>
$x$	: ( $x_1, \dots, x_i, \dots, x_n$ )
$t$	: ( $t_1, \dots, t_k, \dots, t_n$ )
$\delta_k$	: Faktor pengaturan nilai pembobot sambungan pada lapisan <i>output</i>
$\delta_j$	: Faktor pengaturan nilai pembobot sambungan pada lapisan tersembunyi
$\alpha$	: Konstanta laju pelatihan ( <i>learning rate</i> ) $0 < \alpha < 1$
$X_i$	: Unit <i>input</i> ke- $i$ pada lapisan <i>input</i>
$V_{oj}$	: Nilai pembobot pada bias untuk unit $Z_j$
$V_{ij}$	: Nilai pembobot dari unit $X_i$ ke unit $Z_j$
$Z_j$	: Lapisan tersembunyi unit ke- $j$
$Zin_j$	: Net <i>input</i> untuk $Z_j$
$z_j$	: Nilai aktivasi dari unit $Z_j$
$W_{ok}$	: Nilai pembobot pada bias pada <i>output</i> unit $Y_k$
$W_{jk}$	: Nilai pembobot dari $Z_j$ ke unit $Y_k$
$Y_k$	: <i>Output</i> unit ke- $k$ pada lapisan <i>output</i>
$Yin_k$	: Net <i>input</i> untuk $Y_k$
$y_k$	: Nilai aktivasi dari unit $Y_k$
$\Delta W_{jk}$	: Selisih antara $W_{jk}(t)$ dengan $W_{jk}(t+1)$
$\Delta V_{ij}$	: Selisih antara $V_{ij}(t)$ dengan $V_{ij}(t+1)$

$e$  :Error yang terjadi antara *target*  $t$  dan *output*  $y_k$

## 2.6 PID Neural Network [6]

Seperti yang terlihat pada Gambar 2.16, kontroler ini memiliki umpan sederhana jaringan syaraf maju yang terdiri dari 2-3-1 struktur, sehingga memiliki tiga lapisan.



**Gambar 2.16** Struktur PID-Neural Network.

Ada dua *neuron proporsional* pada lapisan *input* dengan mengikuti fungsi aktivasi. Satu untuk menerima pengaturan sistem dan lainnya untuk menerima proses *output*.

$$O_i(k) = \begin{cases} 1 & net_i(k) > 1 \\ net_i(k) - 1 & -1 \leq net_i(k) \leq 1 \\ -1 & net_i(k) < -1 \end{cases} \quad (2.34)$$

Dalam tersembunyi lapisan tiga *neuron* dari berbagai jenis *proporsional*, *integral* dan *derivatif neuron* ada. Fungsi aktivasi untuk *neuron integral* sebagai berikut.

$$O_j(k) = \begin{cases} 1 & O_j(k) > 1 \\ O_j(k-1) + net_j(k) - 1 & -1 \leq O_j(k) \leq 1 \\ -1 & O_j(k) < -1 \end{cases} \quad (2.35)$$

Dan fungsi aktivasi untuk *neuron derivatif* sebagai berikut.

$$O_j(k) = \begin{cases} 1 & O_j(k) > 1 \\ net_j(k) - net_j(k-1) - 1 & -1 \leq O_j(k) \leq 1 \\ -1 & O_j(k) < -1 \end{cases} \quad (2.36)$$

Pada lapisan tersembunyi, *neuron input*

$$net_i = \sum_{j=1}^2 w_{ij} O_j \quad (2.37)$$

**Tabel 2.1** Desain Parameter untuk Skema Berbeda.

Skema	Desain Parameter
PIDNN	$\alpha$ ( <i>learning rate</i> )
Cancellation	$\phi_m$ ( <i>Phase Margin</i> )
Pole Placement	$P$ ( <i>desired pole location</i> )

Pada Persamaan (2.37)  $j=1,2,3,\dots,n$  adalah jumlah *neuron* pada *input* lapisan *neural network*. Akhirnya lapisan tersembunyi terdiri dari satu *neuron proporsional* yang menghasilkan *output controller* sementara *input neuron* adalah sebagai berikut:

$$net_o = \sum_{j=1}^3 w_{oj} O_j \quad (2.38)$$

Dimana  $j$  adalah jumlah *neuron* pada lapisan tersembunyi dan  $o$  adalah *output* lapisan tunggal *neuron*. Belajar dari jaringan ini dilakukan melalui *online* algoritma *backpropagation* fungsi tujuan untuk algoritma ini adalah sebagai berikut dan tujuan dari PIDNN adalah untuk meminimalkan fungsi tujuan ini.

$$J = \frac{1}{N} \sum_{k=1}^N [r(k) - y(k)]^2 \quad (2.39)$$

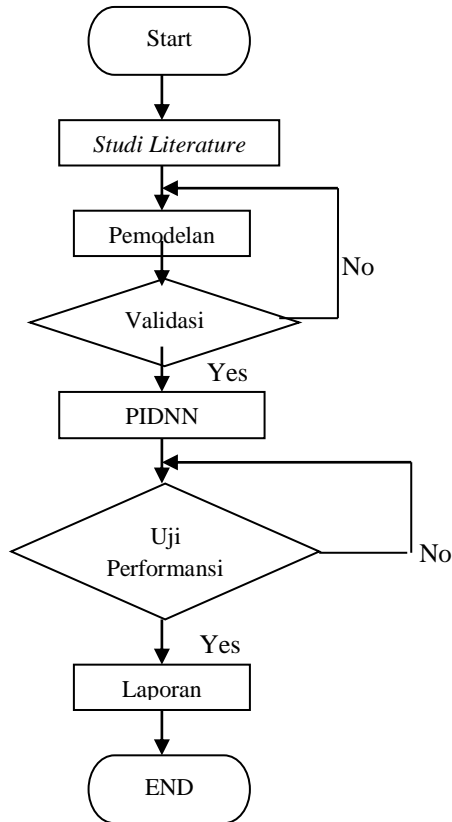
Di mana  $N$  adalah jumlah total *interval sampling*.

### BAB III

#### PERANCANGAN SISTEM

Pada bab III dibahas tentang perancangan sistem secara detail berupa identifikasi kebutuhan untuk sistem, identifikasi sistem, perancangan dan pemodelan *plant* serta perancangan kontroler berisi tentang langkah pembuatan kontroler PID-*Neural network* pada matlab.

#### 3.1 Diagram Alir Penelitian



**Gambar 3.1** Diagram Alir Penelitian.

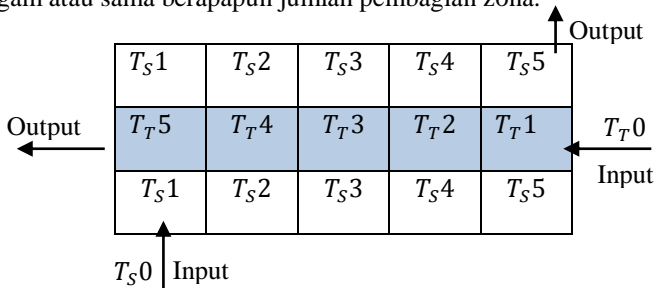
### 3.2 Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan sistem yang digunakan untuk memprediksi kebutuhan yang akan digunakan untuk penelitian suatu tugas akhir. tahap ini sangat penting karena keberhasilan kontrol didalam *plant* sangat bergantung pada komponen penyusun (*sub system*) didalamnya dimana pemilihan suatu komponen didasarkan pada data yang jelas dan lengkap agar tidak terjadi kesalahan yang dapat berakibat gagalnya suatu penelitian.

### 3.3 Perancangan dan Pemodelan *Heat Exchanger*

Dalam tugas akhir ini, *plant* yang dianalisa merupakan *plant* yang didapat dari *paper*. Sehingga untuk merancang suatu sistem yang dapat dikendalikan, maka terlebih dahulu mengidentifikasi *plant* yang digunakan sehingga dapat diketahui karakteristik dari suatu *plant* tersebut. Untuk mendukung pembuatan sistem maka diperolehkannya pemodelan matematis yang diperoleh pada sebuah *paper* dengan judul "*Modelling of Dynamic Countercurrent Tubular Heat Exchanger*" dituliskan dalam sebuah *paper* tersebut mengenai bentuk *nonlinear* suatu sistem *heat exchanger* dengan proses.

Proses identifikasi yang dilakukan yaitu dengan menggunakan metode *open loop* secara simulasi dengan bantuan matlab yang didapat dari suatu persamaan *plant*. Proses selanjutnya dengan melakukan identifikasi dengan mencari nilai masukkan yang berupa sinyal step agar tetap bekerja pada daerah yang ditentukan atau wilayah kerjanya (*operating area*). Untuk memperoleh hasil yang maksimal *heat exchanger* dibagi menjadi N bagian yang sama besar, pada tugas akhir ini dibagi menjadi 5 bagian (Gambar 3.2) dengan panjang perbagian 1,80 meter. Dengan asumsi bahwa temperatur pada satu zona akan seragam atau sama berapapun jumlah pembagian zona.



**Gambar 3.2** Pembagian Zona *Heat Exchanger*.

Untuk mendapatkan persamaan dari setiap zona *heat exchanger* seperti pada Gambar 3.2 adalah dengan menggunakan rumus kesetimbangan energi seperti dibawah ini.

$$\frac{de}{dt} = H_{in} - H_{out} + Q(t) \quad (2.40)$$

Di mana  $\frac{de}{dt}$  adalah laju perubahan terhadap waktu dari energi didalam sistem,  $H_{in}$  adalah energi *flow* yang masuk,  $H_{out}$  adalah energi *flow* yang keluar pada sistem dan  $Q(t)$  adalah *heat transfer* dari sistem.

Dengan asumsi bahwa *heat exchanger* terisolasi dengan baik *plant*, panas yang tidak penting hilang disekitarnya. Dengan menggunakan hubungan termodinamika :

$$\frac{de}{dt} = \rho V C_p \frac{dT_{pout}}{dt} \quad (2.41)$$

Di mana  $\rho$  adalah kepadatan cairan produk, dalam kg/lt,  $V$  adalah volume dari penukar panas,  $C_p$  adalah kapasitas panas dari cairan produk, dalam kcal/kg dan  $\frac{dT_{pout}}{dt}$  adalah laju perubahan temperatur pada *fluida*. Dimana persamaannya sebagai berikut:

$$H_{in} = \rho C_p F_i (T_{pin} - T_{ref}) \quad (2.42)$$

$$H_{out} = \rho C_p F_i (T_{pout}(t) - T_{ref}) \quad (2.43)$$

Di mana  $H_{in}$  adalah energi *flow* yang masuk dalam sistem,  $H_{out}$  adalah energi *flow*,  $F_i$  adalah *flow* yang masuk ke dalam *heat exchanger*,  $T_{pin}$  adalah temperatur *input fluida*,  $T_{pout}(t)$  adalah temperature *fluida output*, dan  $T_{ref}$  adalah temperatur *reference*. Dari penurunan rumus diatas, didapatkan rumus perpindahan panas pada *tube and shell* sebagai berikut:

➤ Shell

$$\rho_s V_s C_{ps} \frac{dT_{s,n}}{dt} = F_s C_{ps} (T_{in} - T_{ref}) - F_s C_{ps} (T_{out} - T_{ref}) - Q_n \quad (2.44)$$

➤ Tube

$$\rho_t V_t C_{pt} \frac{dT_{t,n}}{dt} = F_t C_{pt} (T_{in} - T_{ref}) - F_t C_{pt} (T_{out} - T_{ref}) + Q_n \quad (2.45)$$

Sehingga diperoleh bentuk persamaan *nonlinear heat exchanger* yang dibagi menjadi 5 bagian sesuai dengan zona masing-masing adalah sebagai berikut :



$$\begin{aligned} \text{➤ Zona 1} &= N_{(shell,tube)} \\ \frac{dT_{s,1}}{dt} &= Qu. \alpha_1(Ts_0) - Qu. \alpha_1(Ts_1) - \frac{Q_1}{\alpha_2} \end{aligned} \quad (2.46)$$

$$\begin{aligned} \frac{dT_{t,1}}{dt} &= Qa. \beta_1(Tt_0) - Qa. \beta_1(Tt_1) + \frac{Q_5}{\beta_2} \\ \text{➤ Zona 2} &= N_{(shell,tube)} \quad (2.47) \\ \frac{dT_{s,2}}{dt} &= Qu. \alpha_1(Ts_1) - Qu. \alpha_1(Ts_2) - \frac{Q_2}{\alpha_2} \\ \frac{dT_{t,2}}{dt} &= Qa. \beta_1(Tt_1) - Qa. \beta_1(Tt_2) + \frac{Q_4}{\beta_2} \end{aligned} \quad (2.47)$$

$$\begin{aligned} \text{➤ Zona 3} &= N_{(shell,tube)} \\ \frac{dT_{s,3}}{dt} &= Qu. \alpha_1(Ts_2) - Qu. \alpha_1(Ts_3) - \frac{Q_3}{\alpha_2} \\ \frac{dT_{t,3}}{dt} &= Qa. \beta_1(Tt_2) - Qa. \beta_1(Tt_3) + \frac{Q_3}{\beta_2} \end{aligned} \quad (2.48)$$

$$\begin{aligned} \text{➤ Zona 4} &= N_{(shell,tube)} \\ \frac{dT_{s,4}}{dt} &= Qu. \alpha_1(Ts_3) - Qu. \alpha_1(Ts_4) - \frac{Q_4}{\alpha_2} \\ \frac{dT_{t,4}}{dt} &= Qa. \beta_1(Tt_3) - Qa. \beta_1(Tt_4) + \frac{Q_2}{\beta_2} \end{aligned} \quad (2.49)$$

$$\begin{aligned} \text{➤ Zona 5} &= N_{(shell,tube)} \\ \frac{dT_{s,5}}{dt} &= Qu. \alpha_1(Ts_4) - Qu. \alpha_1(Ts_5) - \frac{Q_5}{\alpha_2} \\ \frac{dT_{t,5}}{dt} &= Qa. \beta_1(Tt_4) - Qa. \beta_1(Tt_5) + \frac{Q_1}{\beta_2} \end{aligned} \quad (2.50)$$

Dimana:

$$\begin{aligned} \alpha_1 &= \frac{F_s}{\rho_s V_s} \\ \alpha_2 &= \rho_s V_s C_{ps} \\ \beta_1 &= \frac{F_t}{\rho_t V_t} \\ \beta_2 &= \rho_t V_t C_{pt} \\ Q_1 &= U \cdot A (Ts_1 - Tt_5) \\ Q_2 &= U \cdot A (Ts_2 - Tt_4) \\ Q_3 &= U \cdot A (Ts_3 - Tt_3) \\ Q_4 &= U \cdot A (Ts_4 - Tt_2) \\ Q_5 &= U \cdot A (Ts_5 - Tt_1) \end{aligned}$$

Adapun perhitungan pada setiap zona yang dimulai dari zona 1 sampai zona 5 dapat diperoleh dengan melihat Tabel 3.1 yang merupakan tabel persamaan koefisien pada *plant heat exchanger*.

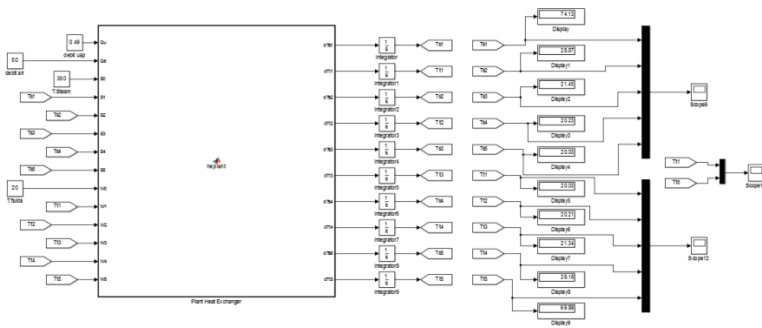
Pada persamaan diatas setiap penentuan nilai dari variabel  $\alpha_1, \alpha_2, \beta_1, \beta_2$  pada persamaan zona 1 sampai zona 5 dapat dicari hasil perhitungannya pada setiap variabel yang dimaksud dengan menggunakan nilai koefisien seperti pada Tabel 3.1 dibawah ini:

**Tabel 3.1** Parameter Koefisien Pada *Plant Heat Exchanger*.

Konstanta	Nilai	Satuan	Keterangan
$U$	10,15	$Kcal / (m^2 min^{\circ}C)$	<i>Overall Heat Transfer</i>
$Dit$	0,02093	$m$	Diameter <i>Internal Tube</i>
$Det$	0,02667	$m$	Diameter <i>Eksternal Tube</i>
$Dis$	0,03505	$m$	Diameter <i>Internal Shell</i>
$L$	9,04	$m$	Total Panjang Pipa
$\rho_s$	1	$Kg/Lt$	Massa Jenis <i>Shell</i>
$C_{ps}$	1	$Kcal/Kg^{\circ}C$	Kapasitas Kalor <i>Shell</i>
$\rho_t$	1	$Kg/Lt$	Massa Jenis <i>Tube</i>
$C_{pt}$	1	$Kcal/Kg^{\circ}C$	Kapasitas Kalor <i>Tube</i>
$\Delta L$	1,808	$m$	Panjang Masing-Masing Pipa
$V_t$	0,0006259	$m^3$	<i>Volume Tube</i>
$V_s$	0,0007267	$m^3$	<i>Volume Shell</i>

Proses identifikasi diatas dilakukan menggunakan metode *open loop* yang akan disimulasikan pada *software* matlab. Tampilan diagram

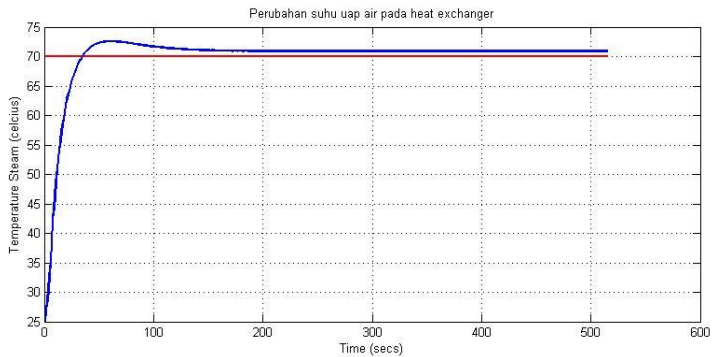
simulasi *open loop plant heat exchanger nonlinear* dapat dilihat pada Gambar 3.3.



**Gambar 3.3** Blok Diagram *Plant Heat Exchanger Nonlinear* pada *Software Matlab*.

Dari Gambar 3.3 diatas diperoleh bahwa temperatur pada suhu *fluida* dingin sebesar  $20^{\circ}\text{C}$ , sedangkan temperatur pada suhu *steam* sebesar  $350^{\circ}\text{C}$ . Nilai  $\alpha_1$  diperoleh dari persamaan  $U/(0,734*1*1)$ ,  $\beta_1$  diperoleh dari persamaan  $U/(0,6217*1*1)$ ,  $\alpha_2$  sebesar  $0,734$ ,  $\beta_2$  sebesar  $0,6217$ , nilai  $U$  sebesar  $10,15$ , dan nilai dari luas penampang sebesar  $\pi * (0,02667/2)^2$ .

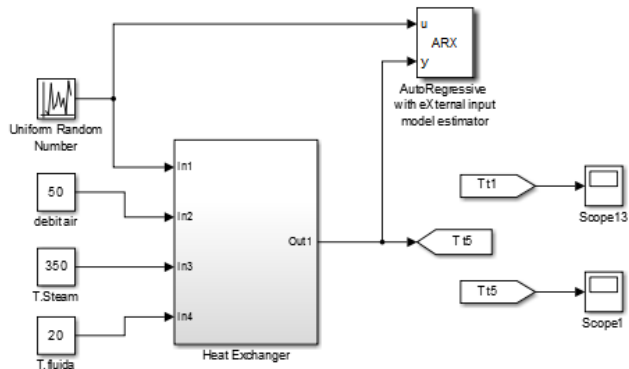
Dengan memberikan nilai kecepatan air dingin sebesar  $50 \text{ m}^2/\text{s}$  dan kecepatan uap air (*steam*) sebesar  $0,5 \text{ m}^2/\text{s}$ , serta memberikan masukkan suhu *fluida* dingin sekitar  $20^{\circ}\text{C}$ , suhu *steam* sekitar  $350^{\circ}\text{C}$ . Maka tampilan simulasi *plant heat exchanger nonlinear* dapat dilihat pada Gambar 3.4 dan waktu simulasi yang diberikan sekitar  $10\text{s}$ , maka diperolehkan *respon open loop*. Mula-mula *respons plant* melonjak hingga mencapai suhu  $72,5^{\circ}\text{C}$ , kemudian turun mendekati sinyal yang terprediksi dan mengalami *steady state* dengan *settling time* pada  $3,2$  detik. Dan temperatur yang menunjukkan *respon* cepat dan mencapai kestabilannya pada nilai  $71,10^{\circ}\text{C}$  dengan *error* sebesar  $1,10\%$ .



**Gambar 3.4** Tampilan Hasil Simulasi *Plant Heat Exchanger Nonlinear* pada *Software Matlab*.

### 3.4 Identifikasi Pemodelan *Plant*

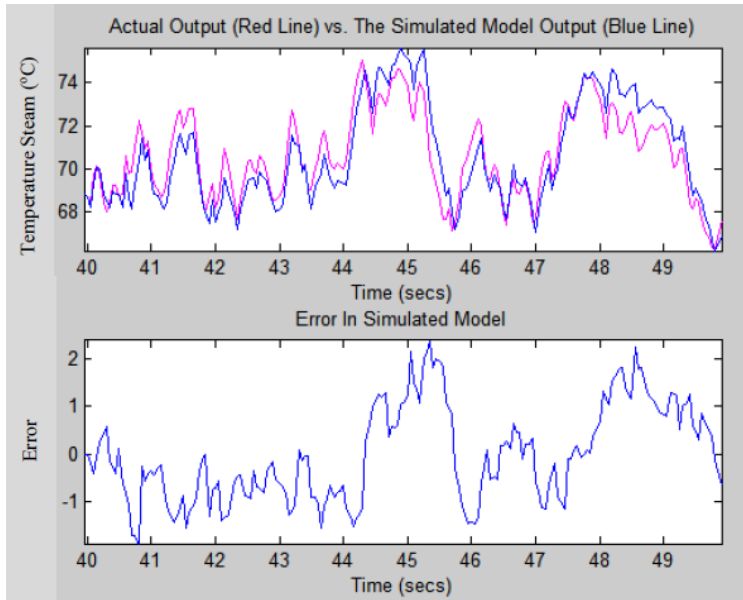
Berdasarkan hasil *respon* persamaan *plant heat exchanger* dapat diketahui *rise time* 0,7s, *time sampling* sekitar 2,87s. Untuk merancang kontroler *PID-Neural Network* diperlukan identifikasi pemodelan *plant heat exchanger* dengan bentuk pendekatan pemodelan ARX dengan pendekatan orde terkecil, namun dengan *error* yang kecil pula. Dalam mencari bentuk ARX dari *plant heat exchanger* digunakan bantuan dari *software matlab* seperti pada Gambar 3.6.



**Gambar 3.5** Tampilan Block Diagram Pendekatan ARX.

Pendekatan ARX dilakukan dengan orde ke-2 masukkan pada *plant*  $50m^2/s$  dan kecepatan air panas pada *plant* yang diberikan berupa nilai

*random* dengan parameter nilai minimum 0,4 dan nilai maksimum 0,6 dan nilai  $n_a=2$ ,  $n_b=1$ ,  $n_k=1$  dengan durasi simulasi 50 dan hasilnya ditunjukkan pada Gambar 3.7 dibawah ini.

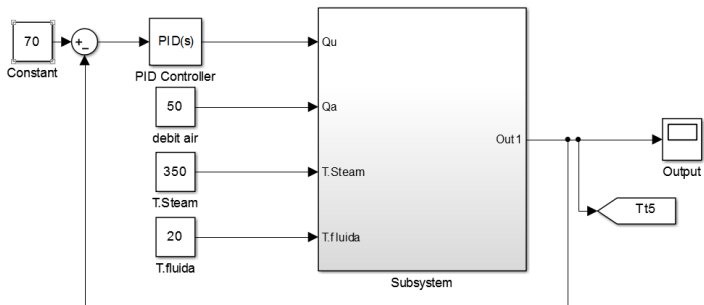


**Gambar 3.6** *Respon Pendekatan ARX Orde ke-2.*

Dari hasil *respon* diatas didapatkan *transfer function*  $\frac{18,367 Z}{Z^2 - 0,57185 Z - 0,29847}$  dengan *time sampling* 0,07 sehingga didapatkan  $a_1=0,57185$  ,  $a_2=0,29847$  , dan  $b_1=18,367$ .

### 3.5 Perancangan Kontroler PID-Neural Network

Setelah melakukan identifikasi *plant*, selanjutnya menentukan nilai awal dari perancangan kontroler *tuning* PID. Untuk mendapatkan nilai *tuning* awal  $K_p$ ,  $K_i$ , dan  $K_d$ , dengan menggunakan blok PID pada simulink matlab. Dimana untuk mendapatkan nilai dari  $K_p$ ,  $K_i$ , dan  $K_d$  adalah dengan menambahkan blok kontroler PID pada *plant heat exchanger*, seperti pada Gambar 3.7.



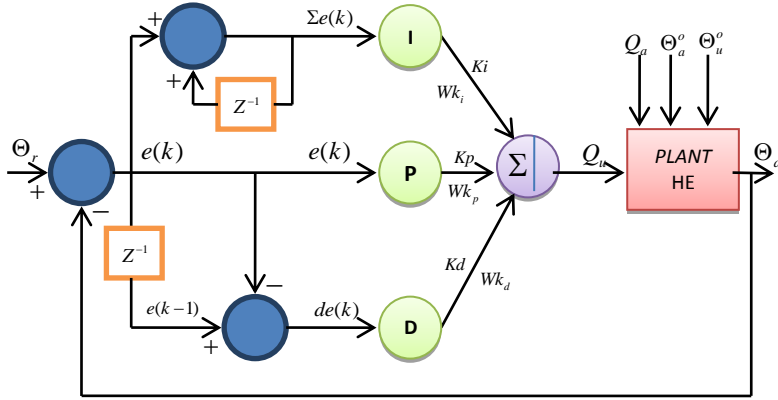
**Gambar 3.7** Blok Diagram untuk Menentukan Nilai Awal  $K_p$ ,  $K_i$ , dan  $K_d$ .

Pada Gambar 3.7 diatas dengan diberikan beban debit air sebesar  $50 \text{ m}^3/\text{s}$  dan temperatur awal *fluida* sebesar  $20^\circ\text{C}$  dan temperatur awal *steam*  $350^\circ\text{C}$ . sehingga didapatkan nilai awal dari  $K_p$ ,  $K_i$ , dan  $K_d$  adalah sebagai berikut:

**Tabel 3.2** Nilai Awal  $K_p$ ,  $K_i$ , dan  $K_d$  untuk Kontroler PID-*Neural Network*.

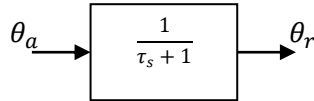
$K_p$	0,0018621425868622
$K_i$	0,0000281777618573199
$K_d$	0,00649728476556897

Setelah didapatkan nilai awal  $K_p$ ,  $K_i$ , dan  $K_d$  dan juga diketahui identitas *plant heat exchanger* maka selanjutnya melakukan perancangan kontroler PID-*Neural network*. Pembuatan kontroler PID-*Neural network* dalam bentuk program dan dengan bantuan m-file dengan dimasukkan nilai awal  $K_p$ ,  $K_i$ , dan  $K_d$  dan nilai dari identitas *plant heat exchanger*. Berikut merupakan blok diagram dari perancangan kontroler PID-*Neural network*.



**Gambar 3.8** Blok Diagram Perancangan Kontroler PID-*Neural Network*.

Dari Gambar 3.8 blok diagram perancangan kontroler PID-*Neural network*, dapat ditentukan estimasi *output* dari kontroler PID-*Neural network* yang dibuat. persamaan estimasi *output* dari blok diagram diatas pada Gambar 3.8 sebagai berikut:



**Gambar 3.9** Blok Diagram Orde 1 dari *Plant Heat Exchanger*.

Dimana *output* dari *plant heat exchanger* merupakan orde 1 dan keluaran dari *plant heat exchanger* dapat mengikuti sinyal *reference* atau sinyal yang diinputkan pada *plant heat exchanger*. Sehingga untuk mendapatkan persamaan estimasi *output* dari kontroler PID-*Neural network* sebagai berikut:

$$\theta_a = \theta_r \left(1 - e^{-\frac{T_s k}{\tau}}\right) \quad (2.51)$$

$$\theta_r = Au(t) \rightarrow \theta_r(s) = \frac{A}{s} \quad (2.52)$$

$$\theta_a = \left(\frac{1}{\tau_s + 1}\right) \left(\frac{A}{s}\right) \rightarrow \theta_a(t) = A \left(1 - e^{-\frac{1}{\tau}t}\right) \quad (2.53)$$

Dari Persamaan (2.53) diubah ke persamaan *domain* (k) dan (k-1) seperti pada Persamaan (2.54) dan (2.55) sebagi berikut:

$$\theta_a(k) = A \left( 1 - e^{-\frac{T_s k}{\tau}} \right) \rightarrow A - A e^{-\frac{T_s k}{\tau}} \quad (2.54)$$

$$\theta_a(k-1) = A \left( 1 - e^{-\frac{T_s (k-1)}{\tau}} \right) \rightarrow A - A e^{-\frac{T_s (k-1)}{\tau}} \quad (2.55)$$

Selanjutnya Persamaan (2.54) dieleminasi dengan Persamaan (2.55) dan menghasilkan persamaan berikut:

$$\theta_a(k) - \theta_a(k-1) = A \left( 1 - e^{-\frac{T_s}{\tau}(k-1)} - e^{-\frac{T_s}{\tau}k} \right) \quad (2.56)$$

Setelah didapatkan hasil eliminasi pada Persamaan (2.56), maka Persamaan (2.51) disederhanakan seperti pada Persamaan (2.57) berikut:

$$\begin{aligned} \theta_a(k) - \theta_a(k-1) &= A \left( e^{-\frac{T_s}{\tau}(k)} \cdot e^{\frac{T_s}{\tau}} - e^{-\frac{T_s}{\tau}k} \right) \\ &= A e^{-\frac{T_s}{\tau}(k)} \left( e^{\frac{T_s}{\tau}} - 1 \right) \end{aligned} \quad (2.57)$$

Dari Persamaan (2.56) didapatkan persamaan sebagai berikut:

$$\theta_a(k) = A - A e^{-\frac{T_s k}{\tau}} \quad (2.58)$$

$$A e^{-\frac{T_s k}{\tau}} = A - \theta_a(k) \quad (2.59)$$

Sehingga persamaan hasil dari eliminasi antara Persamaan (2.54) dan (2.55) sebagai berikut:

$$\theta_a(k) - \theta_a(k-1) = [A - \theta_a(k)] \left[ e^{\frac{T_s}{\tau}} - 1 \right] \quad (2.60)$$

$$\theta_a(k) + \left( e^{\frac{T_s}{\tau}} - 1 \right) \theta_a(k) = \theta_a(k-1) + A \left( e^{\frac{T_s}{\tau}} - 1 \right) \quad (2.61)$$

$$e^{\frac{T_s}{\tau}} \theta_a(k) = \theta_a(k-1) + A \left( e^{\frac{T_s}{\tau}} - 1 \right) \quad (2.62)$$

Sehingga didapatkan persamaan estimasi *output* dari perancangan kontroler PID-*Neural network* sebagai berikut:

$$\theta_a(k) = e^{-\frac{T_s}{\tau}} [\theta_a(k-1) + A \left( e^{\frac{T_s}{\tau}} - 1 \right)] \quad (2.63)$$

Untuk pemrograman kontroler PID-*Neural network*, menggunakan m-file pada *software* matlab. Langkah awal dari pemrograman perancangan kontroler PID-*Neural network* adalah dengan inisialisasi *input* atau masukkan terlebih dahulu yaitu sebagai berikut:

**Tabel 3.3** Inisialisasi *input* pada pemrograman kontroler PID-*Neural network*.

Variabel	Nilai	Keterangan
$Q_u$	0	Debit uap awal
$a_1$	0,57457	Hasil identitas <i>plant</i>
$a_2$	0,29356	Hasil identitas <i>plant</i>
$b_1$	18,367	Hasil identitas <i>plant</i>



Variabel	Nilai	Keterangan
$k_p$	0,0018621425868622	Nilai $k_p$ awal
$k_i$	0,0000281777618573199	Nilai $k_i$ awal
$k_d$	0,00649728476556897	Nilai $k_d$ awal
$\sum e(k)$	0	Penjumlahan bobot <i>error</i>
$e(k - 1)$	0	<i>Error feedback</i>
$\theta(k - 1)$	0	Pemodelan <i>plant</i>
$\theta(k - 2)$	0	Pemodelan <i>plant</i>
$Qu(k - 1)$	0	<i>Output plant</i>
$\tau$	0,3476	<i>Time constant</i>
$t_s$	0,07	<i>Time sampling</i>
$\alpha$	$1 \times 10^{-21}$	<i>Learning gain</i>

Setelah melakukan inisialisasi *input* pada perancangan kontroler PID-*Neural network*, maka diberikan estimasi *output* seperti pada Persamaan (2.58). dan diberikan  $\theta(\text{plant})$  dengan persamaannya sebagai berikut:

$$\theta(\text{plant}) = a1 \cdot \theta(k - 1) + a2 \cdot \theta(k - 2) + b1 \cdot Qu(k - 1) \quad (2.64)$$

Selanjutnya diberikan perhitungan untuk memperbarui nilai dari  $k_p$ ,  $k_i$ ,  $k_d$  seperti pada persamaan berikut:

$$k_p = k_p + \alpha \cdot e(\theta) \cdot e(k) \quad (2.65)$$

$$k_i = k_i + \alpha \cdot e(\theta) \cdot \sum e(k) \quad (2.67)$$

$$k_d = k_d + \alpha \cdot e(\theta) \cdot de(k) \quad (2.68)$$

Dimana:

$$e(\theta) = \theta(\text{plant}) - \text{estimasi output} \quad (2.69)$$

Setelah didapatkan bobot-bobot dari P, I, dan D maka akan di jumlahkan masing-masing bobotnya pada Persamaan (2.64) dan karena *output* dari *plant* tanpa kontroler merupakan *nonlinear* maka diproses dengan aktivasi *linear* seperti pada Persamaan (2.65) dengan *learning* atau *epoch* sebanyak 100 dan persamaan sebagai berikut:

$$Qu = (k_p \cdot e(k)) + (k_i \cdot \sum e(k)) + (k_d \cdot de(k)) \quad (2.70)$$

$$Qu = Qu(k - 1) \quad (2.71)$$

Langkah selanjutnya yaitu melakukan revisi parameter model pendekatan yang bertujuan ketika program m-file mengelola atau memproses untuk mendapatkan nilai dari  $k_p$ ,  $k_i$ , dan  $k_d$  yang sesuai dengan estimasi *output* maka parameter *plant heat exchanger* juga akan menyesuaikan sinyal keluaran dengan sinyal *reference* dengan persamaan seperti berikut:

$$a1 = a1 + \alpha \cdot e(\text{plant}) \cdot \theta(k-1) \quad (2.72)$$

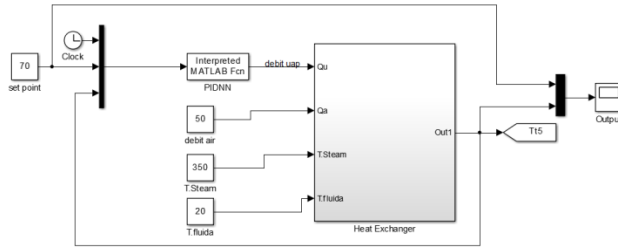
$$a2 = a2 + \alpha \cdot e(\text{plant}) \cdot \theta(k-2) \quad (2.73)$$

$$b1 = b1 + \alpha \cdot e(\text{plant}) \cdot Qu(k-1) \quad (2.74)$$

Dimana  $e(\text{model})$  didapat dari persamaan sebagai berikut:

$$e(\text{plant}) = \theta_a(k) - \theta(\text{plant}) \quad (2.75)$$

Sehingga blok diagram dari perancangan PID-*Neural Network* untuk pengaturan suhu air pada *plant heat exchanger* seperti pada Gambar 3.8 dibawah ini.



**Gambar 3.10** Blok Diagram Perancangan PID-*Neural Network* untuk Pengaturan Suhu Air pada *Plant Heat Exchanger*.

Pada Gambar 3.10 diatas terdapat blok *plant heat exchanger* dan juga blok kontrol PID-*Neural network* dimana pada blok *plant heat exchanger* diberikan masukkan temperatur *steam* sebesar 350 °C, temperatur *fluida* 20 °C, beban debit *fluida* 0,5 m<sup>2</sup>/s dan keluaran dari blok PID-*Neural network* adalah berupa debit *steam*. Dan dengan diharapkan keluaran dari *plant heat exchanger* adalah 70 °C maka diberikan *set point* sebesar 70 °C yang akan diproses terlebih dahulu dengan blok kontrol PID-*Neural network* karena beban yang masuk ke *plant heat exchanger* berubah-berubah. dengan beban yang masuk ke *plant heat exchanger* berubah menyebabkan *error* yang besar, oleh karena itu keluaran dari *plant heat exchanger* akan masuk kembali ke blok kontrol PID-*Neural Network* untuk mendapatkan *error* yang kecil dan keluaran dari *plant heat exchanger* sesuai dengan *set point*.

*Halaman ini sengaja dikosongkan*

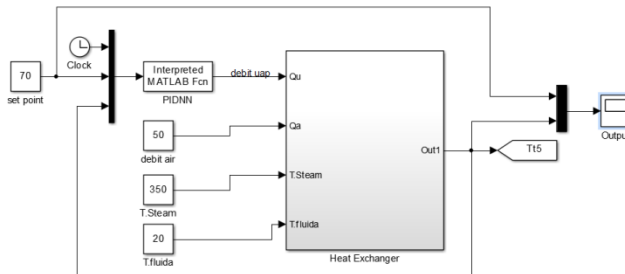
## BAB IV

### PENGUJIAN DAN ANALISA DATA

Berdasarkan perancangan sistem yang telah dibahas pada bab III, langkah berikutnya adalah melakukan simulasi dari perancangan sistem, yang kemudian akan dilakukan analisa terhadap semua hasil simulasi. Uji simulasi dilakukan dengan menggunakan *software* matlab. Tujuan dari uji coba dan analisa adalah untuk mengetahui kinerja dari kontroler PID-*Neural network* tanpa *disturbance* dan dengan *disturbance* dengan parameter yang berbeda-beda. Pada penelitian kali ini penulis membuat 8 keadaan untuk mengetahui seberapa baik *respons plant* yang telah didesain. Dibawah ini akan dijelaskan hasil dari simulasi dengan 8 keadaan yang berbeda-beda untuk mengetahui seberapa baik kontroler yang telah didesain.

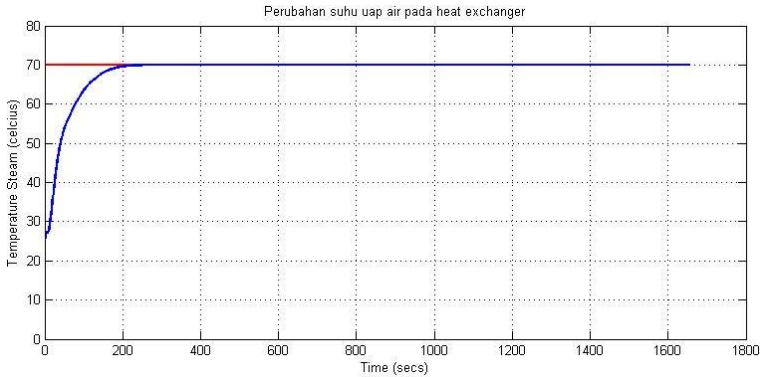
#### 4.1 Set Point 70 °C dengan Beban Konstan

Keadaan pertama nilai dari *set point* sebesar 70 °C dengan nilai beban dibuat konstan sebesar 50 m<sup>2</sup>/s dengan kata lain *plant* tidak terdapat gangguan. Dengan menggunakan metode PID-*Neural Network* diberikan beban konstan dan *set point* juga dibuat konstan, *respons plant* sangat mudah untuk mengikuti model *reference*, karena tidak terdapat gangguan pada *plant* dilihat dari Gambar 4.1 yang merupakan blok diagram dari *set point* sebesar 70 °C dengan nilai beban dibuat konstan sebesar 50 m<sup>2</sup>/s.



**Gambar 4.1** Blok Diagram *Set Point* Konstan Sebesar 70 °C dan Beban Konstan.

Pada Gambar 4.1 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.2 menunjukkan hasil simulink dengan *set point* konstan sebesar 70 °C dengan beban konstan.

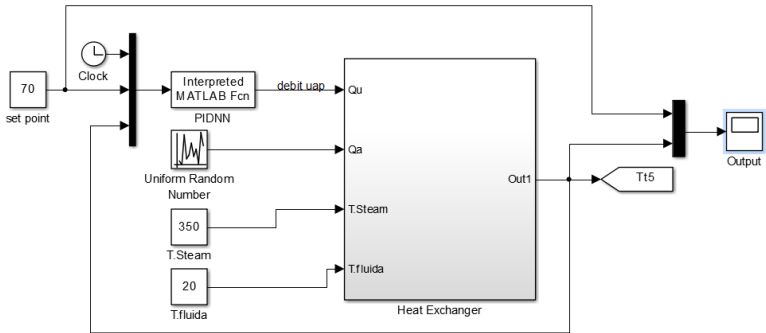


**Gambar 4.2** Hasil Simulink *Set Point* Konstan Sebesar  $70^{\circ}\text{C}$  dengan Beban Konstan.

Dapat dilihat pada Gambar 4.2 bahwa hasil *output* sesuai dengan *set point* sebesar  $70^{\circ}\text{C}$ . Temperatur *fluida* dimulai dari  $25^{\circ}\text{C}$  naik sampai  $70^{\circ}\text{C}$  dan mengalami *steady state* dengan *settling time* pada 4,79s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,4911\text{m}^3/\text{s}$  dan *error steady state* sebesar 0,008%.

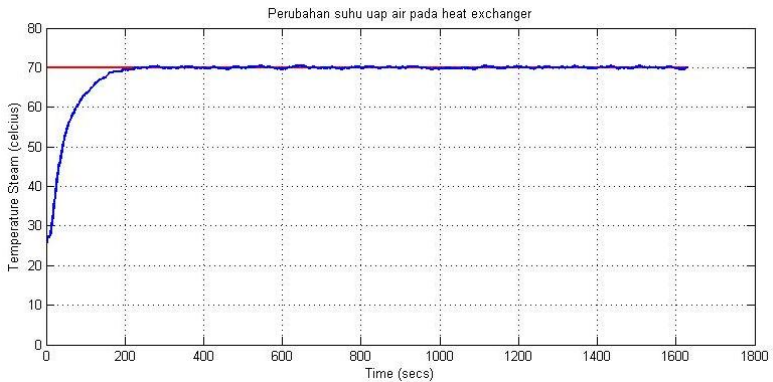
#### 4.2 *Set Point* $70^{\circ}\text{C}$ dengan Beban Bervariasi (*Random*)

Keadaan kedua dengan menentukan *set point* konstan sebesar  $70^{\circ}\text{C}$  dan nilai beban yang dibuat bervariasi antara  $49\text{m}^2/\text{s}$  sampai  $51\text{m}^2/\text{s}$ . Perubahan beban pada *plant heat exchanger* digunakan sebagai gangguan dengan memberikan nilai beban yang bervariasi dengan *range* minimal  $49\text{m}^2/\text{s}$  dan maksimal  $51\text{m}^2/\text{s}$ , hal ini dilakukan dengan tujuan untuk mengetahui sejauh mana kemampuan dari kontroler dapat beradaptasi dengan perubahan beban dari *plant heat exchanger*. Pada Gambar 4.3 ditunjukkan blok diagram dari *set point* sebesar  $70^{\circ}\text{C}$  dengan nilai beban dibuat bervariasi.



**Gambar 4.3** Blok Diagram *Set Point* Konstan Sebesar 70 °C dengan Beban Bervariasi.

Pada Gambar 4.3 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.4 menunjukan hasil simulink dengan *set point* konstan sebesar 70 °C dengan beban bervariasi.

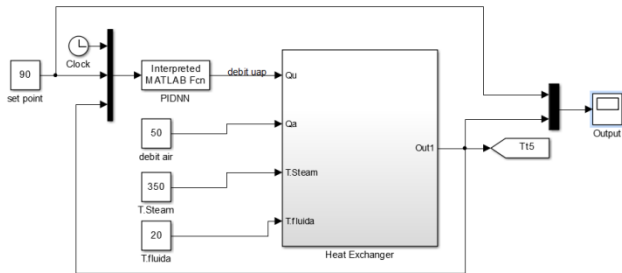


**Gambar 4.4** Hasil Simulink *Set Point* Konstan Sebesar 70 °C dengan Beban Bervariasi.

Dapat dilihat pada Gambar 4.4 bahwa hasil *output* sesuai dengan *set point* sebesar 70 °C. Temperatur *fluida* dimulai dari 25 °C naik sampai 70 °C dan mengalami *steady state* dengan *settling time* pada 4,4s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar 0,4913m<sup>2</sup>/s dan *error steady state* sebesar 0,014%.

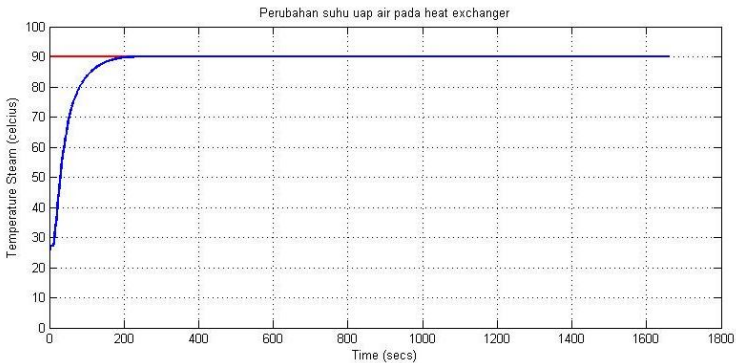
### 4.3 Set Point 90 °C dengan Beban Konstan

Keadaan ketiga nilai dari *set point* sebesar 90 °C dengan nilai beban dibuat konstan sebesar 50 m<sup>2</sup>/s dengan kata lain *plant* tidak terdapat gangguan. Dengan menggunakan metode PID-Neural network diberikan beban konstan dan *set point* juga dibuat konstan, *respons plant* sangat mudah untuk mengikuti *model reference*, karena tidak terdapat gangguan pada *plant* dilihat dari Gambar 4.5 yang merupakan blok diagram dari *set point* sebesar 90 °C dengan nilai beban dibuat konstan sebesar 50 m<sup>2</sup>/s.



**Gambar 4.5** Blok Diagram *Set Point* Konstan Sebesar 90 °C dan Beban Konstan.

Pada Gambar 4.5 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.6 menunjukan hasil simulink dengan *set point* konstan sebesar 90 °C dengan beban konstan.

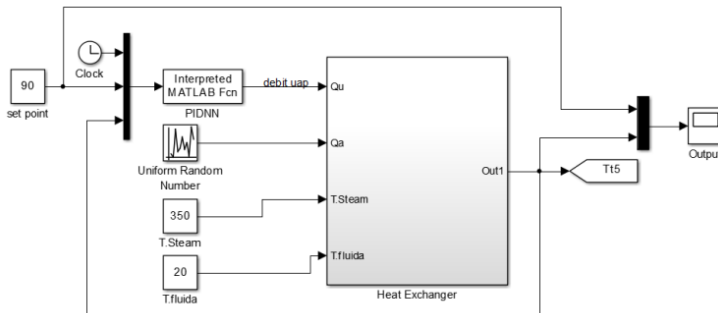


**Gambar 4.6** Hasil Simulink *Set Point* Konstan Sebesar 90 °C dengan Beban Konstan.

Dapat dilihat pada Gambar 4.6 bahwa hasil *output* sesuai dengan *set point* sebesar  $90^{\circ}\text{C}$ . Temperatur *fluida* dimulai dari  $25^{\circ}\text{C}$  naik sampai  $90^{\circ}\text{C}$  dan mengalami *steady state* dengan *settling time* pada 4s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,6878\text{ m}^3/\text{s}$  dan *error steady state* sebesar 0,015%.

#### 4.4 Set Point $90^{\circ}\text{C}$ dengan Beban Bervariasi (*Random*)

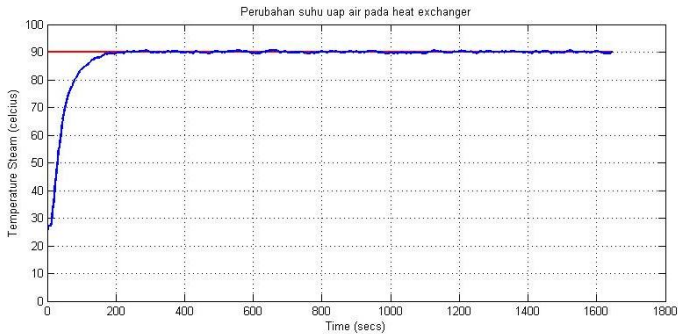
Keadaan keempat dengan menentukan *set point* konstan sebesar  $90^{\circ}\text{C}$  dan nilai beban yang dibuat bervariasi antara  $49\text{ m}^2/\text{s}$  sampai  $51\text{ m}^2/\text{s}$ . Perubahan beban pada *plant heat exchanger* digunakan sebagai gangguan dengan memberikan nilai beban yang bervariasi dengan *range* minimal  $49\text{ m}^2/\text{s}$  dan maksimal  $51\text{ m}^2/\text{s}$ , hal ini dilakukan dengan tujuan untuk mengetahui sejauh mana kemampuan dari kontroler dapat beradaptasi dengan perubahan beban dari *plant heat exchanger*. Pada Gambar 4.7 ditunjukkan blok diagram dari *set point* sebesar  $90^{\circ}\text{C}$  dengan nilai beban dibuat bervariasi.



**Gambar 4.7** Blok Diagram *Set Point* Konstan Sebesar  $90^{\circ}\text{C}$  dengan Beban Bervariasi.

Pada Gambar 4.7 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.8 menunjukan hasil simulink dengan *set point* konstan sebesar  $90^{\circ}\text{C}$  dengan beban bervariasi.



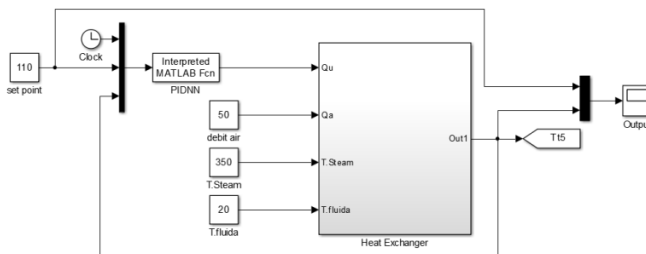


**Gambar 4.8** Hasil Simulink *Set Point* Konstan Sebesar 90 °C dengan Beban Bervariasi.

Dapat dilihat pada Gambar 4.8 bahwa hasil *output* sesuai dengan *set point* sebesar 90 °C. Temperatur *fluida* dimulai dari 25 °C naik sampai 90 °C dan mengalami *steady state* dengan *settling time* pada 4,4s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,6881\text{m}^3/\text{s}$  dan *error steady state* sebesar 0,024%.

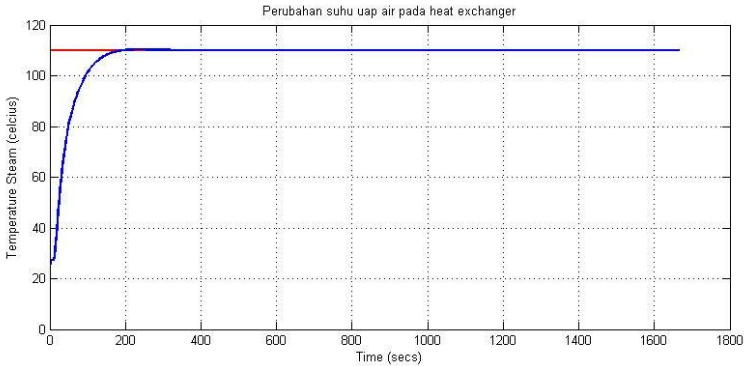
#### 4.5 *Set Point* 110 °C dengan Beban Konstan

Kedadaan kelima nilai dari *set point* sebesar 110 °C dengan nilai beban dibuat konstan sebesar  $50\text{m}^2/\text{s}$  dengan kata lain *plant* tidak terdapat gangguan. Dengan menggunakan metode PID-*Neural network* diberikan beban konstan dan *set point* juga dibuat konstan, *respons plant* sangat mudah untuk mengikuti *model reference*, karena tidak terdapat gangguan pada *plant* dilihat dari Gambar 4.9 yang merupakan blok diagram dari *set point* sebesar 110 °C dengan nilai beban dibuat konstan sebesar  $50\text{m}^2/\text{s}$ .



**Gambar 4.9** Blok Diagram *Set Point* Konstan Sebesar 110 °C dan Beban Konstan.

Pada Gambar 4.9 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.10 menunjukan hasil simulink dengan *set point* konstan sebesar  $110^{\circ}\text{C}$  dan beban konstan.

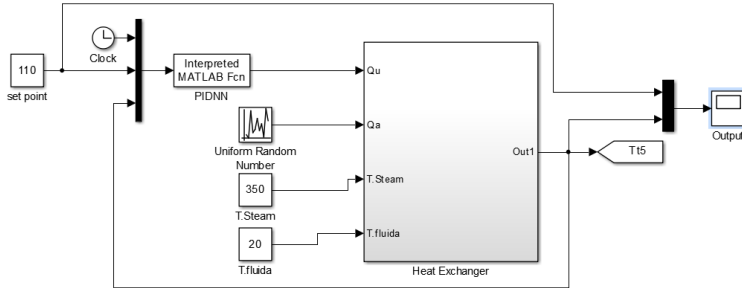


**Gambar 4.10** Hasil Simulink *Set Point* Konstan Sebesar  $110^{\circ}\text{C}$  dan Beban Konstan.

Dapat dilihat pada Gambar 4.10 bahwa hasil *output* sesuai dengan *set point* sebesar  $110^{\circ}\text{C}$ . Temperatur *fluida* dimulai dari  $25^{\circ}\text{C}$  naik sampai  $110^{\circ}\text{C}$  dan mengalami *steady state* dengan *settling time* pada 7,4s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,8853\text{m}^3/\text{s}$  dan *error steady state* sebesar 0,031%.

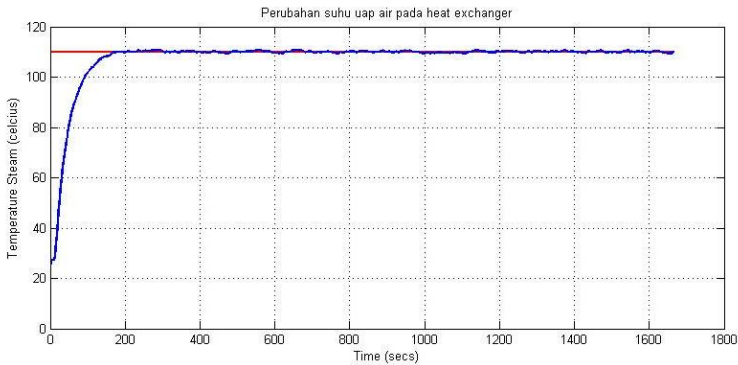
#### 4.6 *Set Point* $110^{\circ}\text{C}$ dengan Beban Bervariasi (*Random*)

Keadaan keenam dengan menentukan *set point* konstan sebesar  $110^{\circ}\text{C}$  dan nilai beban yang dibuat bervariasi antara  $49\text{m}^3/\text{s}$  sampai  $51\text{m}^3/\text{s}$ . Perubahan beban pada *plant heat exchanger* digunakan sebagai gangguan dengan memberikan nilai beban yang bervariasi dengan *range* minimal  $49\text{m}^3/\text{s}$  dan maksimal  $51\text{m}^3/\text{s}$ , hal ini dilakukan dengan tujuan untuk mengetahui sejauh mana kemampuan dari kontroler dapat beradaptasi dengan perubahan beban dari *plant heat exchanger*. Pada Gambar 4.11 ditunjukkan blok diagram dari *set point* sebesar  $110^{\circ}\text{C}$  dengan nilai beban dibuat bervariasi.



**Gambar 4.11** Blok Diagram *Set Point* Konstan Sebesar  $110^{\circ}\text{C}$  dan Beban Bervariasi.

Pada Gambar 4.11 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.12 menunjukan hasil simulink dengan *set point* konstan sebesar  $110^{\circ}\text{C}$  dengan beban bervariasi.

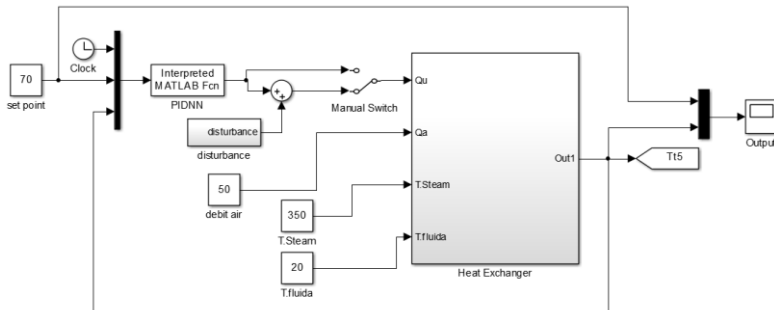


**Gambar 4.12** Hasil Simulink *Set Point* Konstan Sebesar  $110^{\circ}\text{C}$  dan Beban Bervariasi.

Dapat dilihat pada Gambar 4.12 bahwa hasil *output* sesuai dengan *set point* sebesar  $110^{\circ}\text{C}$ . Temperatur *fluida* dimulai dari  $25^{\circ}\text{C}$  naik sampai  $110^{\circ}\text{C}$  dan mengalami *steady state* dengan *settling time* pada 5,6s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,8854\text{m}^3/\text{s}$  dan *error steady state* sebesar 0,034%.

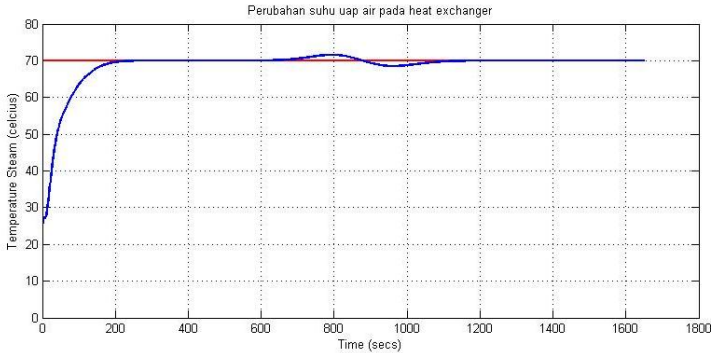
#### 4.7 Set Point 70 °C dengan Beban Konstan dan *Disturbance*

Kedua nilai dari *set point* sebesar 70 °C dengan nilai beban dibuat konstan sebesar 50 m<sup>2</sup>/s dan dengan diberikan *disturbance* atau gangguan pada *plant* pada kondisi 15s. Dengan menggunakan metode PID-Neural network diberikan beban konstan dan *set point* juga dibuat konstan, *respons plant* sangat mudah untuk mengikuti model *reference*, tetapi karena diberikan gangguan atau *disturbance* pada *plant* dengan *magnitude* sebesar 0,05, maka pada waktu 15s terjadi perubahan *set point* tetapi *respons* kembali ke *set point* setelah waktu 15s. Pada Gambar 4.13 yang merupakan blok diagram dari *set point* sebesar 70 °C dengan nilai beban dibuat konstan sebesar 50 m<sup>2</sup>/s dan dengan *disturbance* atau gangguan pada waktu 15s.



**Gambar 4.13** Blok Diagram *Set Point* Konstan Sebesar 70 °C dengan Beban Konstan dan Terdapat *Disturbance* pada *Plant*.

Pada Gambar 4.13 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.14 menunjukan hasil simulink dengan *set point* konstan sebesar 70 °C dengan beban konstan dan *disturbance* atau gangguan pada *plant*.

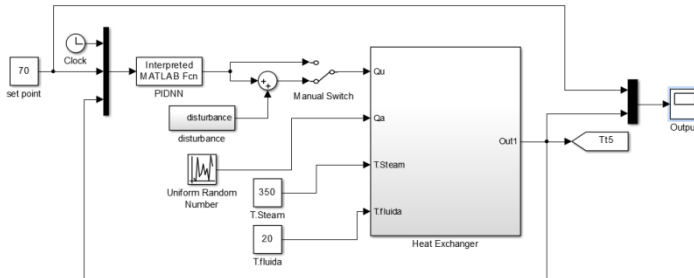


**Gambar 4.14** Hasil Simulink *Set Point* Konstan Sebesar 70 °C dengan Beban Konstan dan *Disturbance* pada *Plant*.

Dapat dilihat pada Gambar 4.14 bahwa hasil *output* sesuai dengan *set point* sebesar 70 °C. Temperatur *fluida* dimulai dari 25 °C naik sampai 70 °C dan mengalami *steady state* dengan *settling time* pada 11,4s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,4911\text{m}^2/\text{s}$  dan *error steady state* sebesar 0,15%.

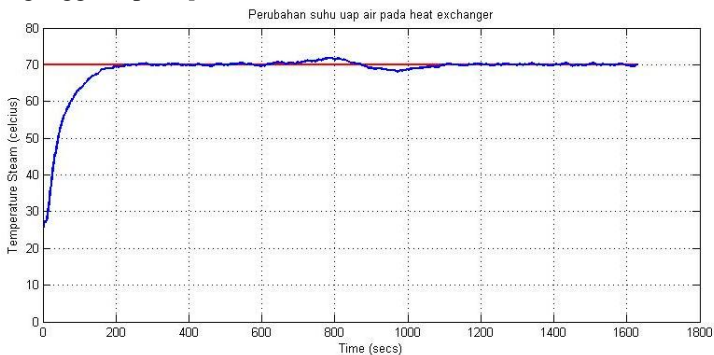
#### 4.8 *Set Point* 70 °C dengan Beban Bervariasi dan *Disturbance*

Keadaan kedelapan nilai dari *set point* sebesar 70 °C dengan nilai beban dibuat bervariasi sebesar  $50\text{ m}^2/\text{s}$  dan dengan diberikan *disturbance* atau gangguan pada *plant* pada kondisi 15s. Dengan menggunakan metode PID-*Neural network* diberikan beban konstan dan *set point* juga dibuat konstan, *respons plant* sangat mudah untuk mengikuti model *reference*, tetapi karena diberikan gangguan atau *disturbance* pada *plant* dengan *magnitude* sebesar 0,05, maka pada waktu 15s terjadi perubahan *set point* tetapi *respons* kembali ke *set point* setelah waktu 15s. Pada Gambar 4.15 yang merupakan blok diagram dari *set point* sebesar 70 °C dengan nilai beban dibuat bervariasi dan dengan *disturbance* atau gangguan pada waktu 15s.



**Gambar 4.15** Blok Diagram *Set Point* Konstan Sebesar 70 °C dengan Beban Bervariasi dan Terdapat *Disturbance* pada *Plant*.

Pada Gambar 4.15 blok diagram diatas dengan diberikan waktu *running* 30s maka pada Gambar 4.16 menunjukan hasil simulink dengan *set point* konstan sebesar 70 °C dengan beban bervariasi dan *disturbance* atau gangguan pada *plant*.



**Gambar 4.16** Hasil Simulink *Set Point* Konstan Sebesar 70 °C dengan Beban Bervariasi dan *Disturbance* pada *Plant*.

Dapat dilihat pada Gambar 4.14 bahwa hasil *output* sesuai dengan *set point* sebesar 70 °C. Temperatur *fluida* dimulai dari 25 °C naik sampai 70 °C dan mengalami *steady state* dengan *settling time* pada 7,7s. Dan pada hasil grafik simulink diatas didapat debit uap sebesar  $0,4913\text{m}^2/\text{s}$  dan *error steady state* sebesar 0,16%.

*Halaman ini sengaja dikosongkan*

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Secara umum dapat dikatakan bahwa sistem kontrol PID-*Neural network* yang didesain untuk mengatur temperatur pada *heat exchanger* dapat bekerja dengan baik dan sudah sesuai dengan kriteria yang diinginkan. Dari uraian yang telah dijabarkan pada bab sebelumnya dapat ditarik kesimpulan sebagai berikut:

1. Kontroler dengan menggunakan metode PID-*Neural network* dapat bekerja dengan baik. Hal tersebut dapat dilihat dari hasil pengujian pada simulink bahwa sinyal *output plant* dapat beradaptasi terhadap *set point* yang berubah-ubah. Pada pengujian dengan beban konstan terjadi *error steady state* sebesar kurang dari 1%. Sedangkan dengan beban bervariasi (*random*) terjadi *error steady state* sebesar kurang dari 1% juga. Dengan *error steady state* kurang dari 1% untuk beban konstan dan untuk beban bervariasi dapat dikatakan bahwa aksi kontroler terhadap perubahan *set point* dan perubahan beban sangat baik.
2. Desain kontroler PID-*Neural network* juga mampu mengikuti perubahan adanya *disturbance* atau adanya gangguan dari luar kontroler maupun *plant*, dan menimbulkan *error steady state* yang mencapai kurang dari 1%. *Error steady state* yang terjadi lebih besar jika dibandingkan dengan mengubah nilai *set point* dan beban yang bervariasi.
3. Lambat atau cepatnya respon *output plant* dapat beradaptasi terhadap perubahan *set point* dipengaruhi oleh besar kecilnya *learning rate* yang digunakan pada struktur jaringan saraf tiruan dan penggunaan bobot-bobot hasil revisi pada proses pemodelan *plant*.

#### **5.2. Saran**

Untuk pengembangan dan penyempurnaan dari perancangan PID-*Neural network* ini agar memiliki kinerja yang lebih baik, maka diberikan beberapa saran sebagai berikut :

1. Saran untuk penelitian selanjutnya diharapkan dilakukan dengan menggunakan *real plant* dengan proses identifikasi terlebih dahulu sedangkan dari segi kontrolnya dapat ditambahkan kontrol adaptif agar kontroler dapat beradaptasi dengan kondisi



apapun dan sehingga *output plant* dapat mengikuti perubahan beban dan dapat mengurangi *error* pada saat *tracking*.

## DAFTAR PUSTAKA

- [1] Belinda Chong, Mohd Nor B, 2010. Modelling of A Hot Water Drum and Heat Exchanger Process Control Training System. *Journal of IEEE International Conference on Control, Automation, Robotics, and Vision*. vol. 7 (10): 1489-1494.
- [2] Abdulrahman, Rosbin, Hanafi, 2012. Mathematical Modeling of Industrial Heat Exchanger System. *Journal of Aplied Mechanics and Materials on Trans Tech Publications Switzerland*. vol.229-231: 2122-2124.
- [3] Rahmawati,R.r. 2012. Pengendali Temperatur Fluida pada Heat Exchanger Dengan Menggnakan Jaringan Syaraf Tiruan Prediktif.*Tugas Akhir*. Teknik Elektro.Surabaya: Institut Teknologi Sepuluh Nopember.
- [4] Mithun, Satheeshbabu.R. 2015. Nonlinear Controller Design for A Shell and Tube Heat Exchanger. *International Journal of Electrical, Electronics and Data Communication*. vol.3 : 2320-2084.
- [5] Ekasari, R. P., 2013. Pengatur Temperatur Pada Heat Exchanger Dengan Menggunakan Metode Jaringan Syaraf Tiruan Prediktif. *Tugas Akhir*. Teknik Elektro.Surabaya: Institut Teknologi Sepuluh Nopember.

*Halaman ini sengaja dikosongkan*

# LAMPIRAN

## LAMPIRAN A

### *Listing Program*

#### **Program *plant heat exchanger***

```
function
[dTs1,dTt1,dTs2,dTt2,dTs3,dTt3,dTs4,dTt4,dTs5,dT
t5] = heplant(Qu, Qa, S0 ,S1 ,S2 ,S3 ,S4 ,S5 ,W0
,W1 ,W2 ,W3 ,W4 ,W5)
%#codegen
alp1=0.62;
alp=2.7564;
bta1=0.73;
bta=0.15172;
U=10.15;
A=1.71251*3.14;

%shell
dTs1=Qu*alp*(S0)-Qu*alp*(S1)-(U*A*(S1-W5))/alp1;
dTs2=Qu*alp*(S1)-Qu*alp*(S2)-(U*A*(S2-W4))/alp1;
dTs3=Qu*alp*(S2)-Qu*alp*(S3)-(U*A*(S3-W3))/alp1;
dTs4=Qu*alp*(S3)-Qu*alp*(S4)-(U*A*(S4-W2))/alp1;
dTs5=Qu*alp*(S4)-Qu*alp*(S5)-(U*A*(S5-W1))/alp1;
%tube
dTt1=Qa*bta*(W0)-Qa*bta*(W1)+(U*A*(S5-W1))/bta1;
dTt2=Qa*bta*(W1)-Qa*bta*(W2)+(U*A*(S4-W2))/bta1;
dTt3=Qa*bta*(W2)-Qa*bta*(W3)+(U*A*(S3-W3))/bta1;
dTt4=Qa*bta*(W3)-Qa*bta*(W4)+(U*A*(S2-W4))/bta1;
dTt5=Qa*bta*(W4)-Qa*bta*(W5)+(U*A*(S1-W5))/bta1;
end
```

## Program PID-*Neural Network*

```
function Output = pidnnk(xin)
%#codegen
global erk ekml jek kp ki kd thk thkm1 thkm2 ts
t
global alph erth dek al a2 b1 thkmd Qukm1 thes
Qu
tt=xin(1);
thr=xin(2);
thk=xin(3);
erk=thr-thk

%inisialisasi
if tt==0
    Qu=0;
    a1=0.57457;
    a2=0.29356;
    b1=18.367;
    kp=0.0018621425868622;
    ki=0.0000281777618573199;
    kd=0.00649728476556897;
    jek=0;
    ekml=0;
    thkm1=0;
    thkm2=0;
    Qukm1=0;
    t=0.3476;
    ts=0.07;
    alph=0.0000000000000000000001; %learning
gain
end
jek=jek+erk;
dek=erk-ekml;

if tt< 2*ts
    epp=0;
else
```

```

epp=100;
for i=1:epp
    thes=(exp(-ts/t))*(thkm1+thr*(exp(ts/t)-
1)); %estimasi output
    thkmd=a1*thkm1+a2*thkm2+b1*Qukm1;
    erth=thkmd-thes;

    kp=kp+alph*erth*erk;
    ki=ki+alph*erth*jek;
    kd=kd+alph*erth*dek;

    Qu=(kp*erk)+(ki*jek)+(kd*dek);
    Qukm1=Qu;
end
end

thkmd=a1*thkm1+a2*thkm2+b1*Qukm1;
ermd=thk-thkmd;

%revisi parameter model pendekatan
a1=a1+alph*ermd*thkm1;
a2=a2+alph*ermd*thkm2;
b1=b1+alph*ermd*Qukm1;

ekm1=erk;
thkm2=thkm1;
thkm1=thk;

Output=Qu

```

*Halaman ini sengaja dikosongkan*

## **RIWAYAT PENULIS**



Arie imanda lahir di purwakarta, jawa barat pada tanggal 5 januari 1994. Penulis memulai pendidikannya dari TK dharmawanita di Sidoarjo, kemudian melanjutkan studinya di SDN Wadungasih 2 Buduran, Sidoarjo, SMPN 1 Buduran, dan SMA Muhammadiyah 2 Sidoarjo. Setelah menamatkan SMA, penulis melanjutkan studinya di D3 Teknik Elektro FTI-ITS dan lulus pada tahun 2015. Selanjutnya penulis meneruskan studi sarjana di Teknik Elektro FTE-ITS, kemudian fokus pada bidang studi Teknik Sistem Pengaturan. Pada bulan Juli 2017, penulis mengikuti seminar dan Ujian Tugas Akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro dari Institut Teknologi Sepuluh Nopember Surabaya.



*Halaman ini sengaja dikosongkan*